# DOCUMENT RESUME

ED 230 181 IR 010 702

| | |
|---|---|
| TITLE | Computer Literacy. Intelligent CAI. Proceedings of the 1982 Annual Conference of the Association for the Development of Computer-Based Instructional Systems. |
| INSTITUTION | Association for the Development of Computer-based Instructional Systems. |
| PUB DATE | 82 |
| NOTE | 387p. |
| PUB TYPE | Collected Works - Conference Proceedings (021) -- Viewpoints (120) -- Reports - Descriptive (141) |
| EDRS PRICE | MF01/PC16 Plus Postage. |
| DESCRIPTORS | Audiovisual Aids; *Computer Assisted Instruction; Computer Graphics; *Computer Literacy; *Computer Oriented Programs; Computers; Electronic Equipment; Material Development; Media Research; Production Techniques; *Program Descriptions; Simulation; *Training Methods |
| IDENTIFIERS | *Computer Uses in Education; Intelligent Videodisc Systems; PLATO |

ABSTRACT

This collection of 116 papers begins with 5 presentations on the general theme of computer literacy and intelligent computer assisted instruction, and covers a wide variety of topics related to training and education at all grade levels. The remaining papers are divided according to special interest group categories: (1) computer-based training; (2) elementary, secondary, junior college; (3) educators of the handicapped; (4) health education; (5) computer-based home economic instruction; (6) implementation; (7) math; (8) computer-based music; (9) PLATO users; (10) mini/microcomputer systems in computer-based instruction; (11) theory and research; and (12) project reports. Within the broad topic categories, individual papers discuss simulation, computer graphics, computer software, hardware, production systems, computer literacy, specific projects, and case studies. Most papers include abstracts and references. (LMM)

***********************************************************************
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
***********************************************************************

ED230181

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

- This document has been reproduced as received from the person or organization originating it.
- ☐ Minor changes have been made to improve reproduction quality.

- Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

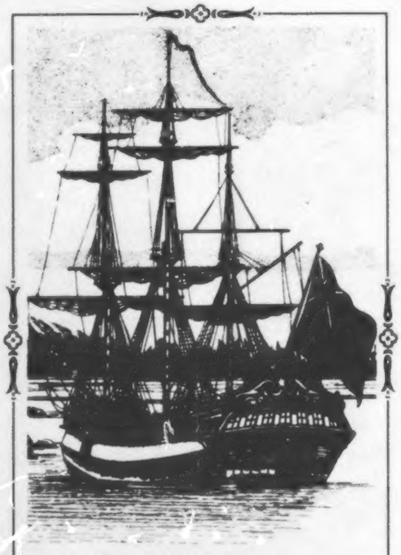
ADCIS 82:
COMPUTER
LITERACY,
INTELLIGENT C..I

# 1982
# CONFERENCE PROCEEDINGS

COMPUTER LITERACY

INTELLIGENT CAI

# Association for the Development of Computer-Based Instructional Systems



CAPTAIN COOK'S SHIP
RESOLUTION AT ANCHOR
IN FRIENDLY COVE IN 1778

"PERMISSION TO REPRODUCE THIS
MATERIAL HAS BEEN GRANTED BY
Gordon Hayes

TO THE EDUCATIONAL RESOURCES
INFORMATION CENTER (ERIC)."

Copyright © ADCIS 1982
Western Washington University
Bellingham, Washington 98225
Printed in U.S.A.

IR010702

# ADCIS Board of Directors

***President***
Michael Allen
Control Data Corporation

***Secretary-Treasurer***
Rita Wagstaff
Cuyahoga Community College

***Program Chairperson***
Emile Attala
California Polytechnic State University

***JCBI Editor***
Keith Hall
The Ohio State University

***Chapter Representative***
Charles Buchanan
United Airlines

***Special Interest Group Representative***
Ron Comer
OSU College of Medicine

***Special Interest Group Representative***
G. David Peters
University of Illinois

***Parliamentarian & Governance Chairperson***
Joan Hayes
Western Washington University

***ADCIS Executive Secretary***
Gordon Hayes
ADCIS International Headquarters

## Special Interest Group Council

***Computer-Based Training***
Robert C. Fratini
Western Electric

***Elementary / Secondary / Jr. College***
Daryl R. VanAlstyne
Boces Monroe

***Educators for the Handicapped***
Jan Richardson
Gallaudet College

***Health Education***
Ron Comer
The Ohio State University

***Home Economics***
Fran Mayhew
University of Delaware

***Implementation***
Edward Sallin
South Central Bell

***Math***
Ron Wenger
University of Delaware

***Mini/Micro***
Bob Tannenbaum
Hunter College

***Music***
G. David Peters
University of Illinois

***PLATO***
Larry Francis
Battelle Labs

***Theory and Research***
Gary Feurer
University of Delaware

Frank Roberts
Control Data Corporation

## Charters Council

Chuck Buchanan
FRONT RANGE CHAPTER

Rick Anderson
OHIO CHAPTER

Sally Coppus
NORTHEAST CHAPTER

# FOREWORD

In this turning point in our lives where the information revolution is replacing energy as the key to productivity growth, Computer-based educational and training systems are economic realities which have begun to transform the world in which we live. Consequently, we must update our teaching modes by using more sophisticated equipment to allow the students to take a more active role in their own education. The expansion and refinement of computer technology has given us the opportunity to design these educational models which seek to find the best union between man and machine.

Parallel with the advances in computer technology we must move with increasing speed to provide computer literacy to the public in general and to educators in particular. For CAI to be effective, we as educators must be equipped to adapt to the changes imposed on us by the new technologies so that we can help our students accept CAI as a mode of instruction and use it properly. This is the challenge and we must meet it in this decade.

The papers contained in this volume represent much of the thinking about Computer Literacy and Intelligent CAI. By choosing these two subjects as the theme of the 1982 Conference, ADCIS, via its dedicated membership, is meeting the challenge of providing valuable information to its audience.

I would like to express my thanks for the assistance and support provided by the SIG chairpersons of ADCIS. I am grateful to my secretary, Lynda Alamo, for typing countless pages, for her infinite patience, and for unending support. I want also to thank Peggy Bishop from our Headquarters Office who played a major role in streamlining the interface between the contributors of this conference. In particular, I would like to express my gratitude to Gordon Hayes, Executive Secretary, for his understanding, support and encouragement. It is indeed a pleasure to acknowledge my debt for his help.

***Emile Attala***
***1982 Program Chairperson***

# TABLE OF CONTENTS

## GENERAL TOPIC PRESENTATION

Aut Vincere Aut Mori ........ 1
*Jans Diemer*

Evaluation of Computer-Based Instruction Systems ........ 5
*Donald L. Shirer*

Designing Languages for Micros ........ 12
*George Gerhold, Larry Kheriaty*

Computer-Assisted Education: What's Not Happening? ........ 16
*Francis D. Fisher*

Software — How Good Is It? ........ 22
*Florence M. Taber, Alonzo E. Hannaford*

## COMPUTER-BASED TRAINING SPECIAL INTEREST GROUP PAPERS

Computer Literacy in Business and Industry: Experiences With Microcomputer Workshops ........ 25
*Greg Kearsley*

On Software Training Via CAI: Some Problems and Possible Solutions ........ 31
*Michael J. Zimmerman*

Year 7 - Federal Aviation Administration: CBT Status Report ........ 34
*John A. Buck*

Project U-Train CAL: Authoring Methods for Subject Matter Experts Using Natal-II ........ 37
*Griff Richards, Jean Belanger*

Computer Training: From Secretarial to Technical Personnel ........ 42
*John Green, Enid Squire*

Hand-Held Computers — For CBI? ........ 47
*Larry Francis*

Integrating Simulation Activities with Instruction: Guidelines for Developers ........ 52
*David R. Abernathy, Kathleen A. Terry*

Computer-Based Training (CBT): Buying Criteria ........ 57
*W. J. Hudson*

Graphics Simulation in Maintenance Training - Training Effectivenes at Cost Savings ........ 62
*Brian Dallman*

An Integrated Design and Development System for Graphics Simulation ........ 65
*J. Jeffrey Richardson*

Graphics Simulation System Run Time Operation ........ 70
*William Pieper*

An Editor for the Automated Generation of Graphics Simulations ........ 74
*Anne Montgomery*

Graphics Simulation Hardware - A Twin Display System Utilizing a Video Disc and a Medium-Resolution Color Graphics Terminal ........ 79
*Dale Steffen*

Relationship of Performance on a Computer-Based Context-Free Fault Diagnosis Simulation to First-Semester Performance in the Aviation Mechanics Curriculum .......... 85
*Karen S. Gaddis, William B. Johnson, Stanley R. Trollip*

## ELEMENTARY / SECONDARY / JUNIOR COLLEGE SPECIAL INTEREST GROUP

Faculty Computer Literacy: Five Strategies .......... 89
*Alan Creutz*

Computer Literacy for Teachers: A District Do-It-Yourself Kit .......... 93
*Glenn Bruckhart, Judith A. Hopper*

Microcomputers and Achievement Monitoring: A Partial Solution to the Demands for Educational Efficiency and Accountability .......... 97
*Charles G. Boody*

The Affective Response of Students to CAI .......... 104
*William E. Russell*

The Computer as a Component in a Multi-Method Approach to Teaching Anatomy and Physiology .......... 109
*Donald A. Wheeler, Mary Jane Burge*

The Use of Production Rules in the Development of an Expert Tutor for Algebra .......... 113
*William S. Bregar, Arthur M. Farley, Brian Lantz*

## EDUCATORS OF THE HANDICAPPED SPECIAL INTEREST GROUP

Microcomputers and the Handicapped .......... 122
*Chris Dumper*

## HEALTH EDUCATION SPECIAL INTEREST GROUP

Creating CAI for a Clinical Laboratory .......... 126
*Tracie M. Jenkins*

Toward Natural Language Authoring of Lessons on PLATO .......... 132
*K. M. Bagnall, M. Szabo, I Rollingher, A. Kallal*

The CAI Microcomputer Project in the Faculty of Nursing at the University of Calgary .......... 136
*K. J. Hannah, D. N. Conklin*

The Use of Computer Simulation to Teach Calculation and Regulation of Intravenous Flow Rates to Nursing Students .......... 141
*Donna E. Larson*

The Use of Nursing Process in Computer-Based Simulation Design .......... 148
*Kathleen D. Kadner*

CAI and Instructional Interactions .......... 151
*Louis A. Pagliaro*

## COMPUTER-BASED HOME ECONOMICS INSTRUCTION SPECIAL INTEREST GROUP

Computer-Managed Learning as Facilitator of Individualized Learning in Textile Science .......... 161
*E. M. Crown, Maryanne Doherty*

National Survey for Computer-Based Educational Software in Home Economics .......... 166
*Frances Whitaker Mayhew*

Simulated Laboratory Experiments Investigating Functional Properties of Foods .......... 170
*Zoe Ann Holmes*

## IMPLEMENTATION SPECIAL INTEREST GROUP

Student Edited Graphics .................................................... 174
*Thomas J. Chapin, Richard W. Davis*

An Implementation Structure for Instructional Computing .................................... 179
*Eileen K. Gress*

MENO-II: An AI-CAI Programming Tutor .................................................... 185
*Elliot Soloway, Eric Rubin, Beverly Woolf, Jeffrey Bonar*

Courseware Transportability Issues .................................................... 193
*Richard C. Brandt*

## MATH SPECIAL INTEREST GROUP

The "Buggy Metaphor": Themes in Algebra and Precalculus .................................... 197
*Ronald H. Wenger*

But I Don't Want to Know All About .................................................... 203
*J. Arthur Seebach*

Issues in the Development of Computer Graphics for Mathematics Instruction .................... 210
*Mark J. Christensen*

Text Materials for Computer Literacy .................................................... 212
*Sarah Brooks*

Production Systems for Computer-Based Learning .................................................... 213
*Alfred Bork*

Panel on Software Production Systems .................................................... 217
*David A. Smith*

Computer-Assisted Discovery in the Mathematics Classroom .................................... 218
*Marjorie A. Fitting*

Impact of Computers on Precollege Mathematics: The Present and the Future .................... 219
*Jerry Johnson, Tony Jongejan, Cheryl Lemke*

The Swarthmore Computer Trigonometry Project .................................................... 220
*Eugene Klotz*

## COMPUTER-BASED MUSIC SPECIAL INTEREST GROUP

Music Terminology of Children, College Students and Faculty: Applications for Computer-Assisted Lessons .................................................... 222
*Harriet I. Hair*

The Use of a Table Driver for Individualized Design of Computer-Based Instruction Materials in Music Theory .................................................... 228
*Michael Arenson*

## PLATO USERS' SPECIAL INTEREST GROUP

A Pilot Study of a Spanish Version of the PLATO Basic Skills Learning System Mathematics Curriculum .................................................... 231
*Patricia M. Mullinix, Ralph T. Heimer*

## MINI/MICRO COMPUTER SYSTEMS IN CBI SPECIAL INTEREST GROUP

Networked Microcomputers Problems and Prospects in University Use .................................... 235
*Paul Lorton, Jr., John Eaton*

Towards Portable Microcomputer Courseware ........ 239
*Norman F. Salt*

The CLEF Project: Learning French on Colour Micros ........ 245
*Glyn Holmes, Marilyn Kidd*

Cooperative Courseware Production: A Solution to the Problems of CAI? ........ 252
*Glynn Holmes, Donna Mydlarski, Dana Paramskas*

The Flic Project: Micros and Minis ........ 259
*Donna Mydlarski, Ann Brebner, H. J. Hallworth*

Design and Implementation of a Micro-based CAI Course for a Microfiche Processor ........ 263
*James A. Cook*

An Apple Computer as a Terminal for a Unix-Based Educational Network ........ 267
*Gareth R. Shearman*

Computer Literacy — Keeping the Options Open ........ 270
*Annalee Elman*

Coordination and Evaluation of Videodisc and Microcomputer Programs Across 45 Sites ........ 276
*Donald H. McLaughlin*

## THEORY AND RESEARCH SPECIAL INTEREST GROUP

Intrusive and Non-Intrusive Instruction in Dynamic Skill Training ........ 283
*Allen Munro, Michael R. Fehling, Douglas M. Towne*

An Instructor of Associations Which Learns ........ 289
*M. J. Chas. E. Woodson*

A Trip to Europe . . . Apparel Terminology in Six Foreign Languages ........ 292
*Karen E. Schaeffer*

Computer-Based Instruction of Expert Problem Solving Procedures: Ensuring Access to Relevant Knowledge Structures ........ 298
*Richard W. Evans*

Forced Correction in Coordinate Concept Learning ........ 306
*Timothy W. Spannaus*

# LATE PAPERS

The Microcomputer, Mastery Learning, and the Changing Role of the Instructor ........ 311
*Chris Dimas*

## Project Reports

CAI in Foreign Language Teaching: Components of a Microcomputer Station ........ 317
*Robert A. Ariew*

An Anatomy Self-Testing Program on PLATO ........ 318
*K. M. Bagnall, M. Szabo, R. Frost, I. Rollingher*

Integrated Music Project ........ 320
*W. Kenton Bales, Kenneth N. Paoli, Elizabeth Weherman*

Computer Literacy: A Survey of Teachers K Through 12 ........ 321
*Julie Bengston*

Project U-Train/CAL — Three Years On ........ 322
*J. J. R. Belanger, Griff Richards*

Office of Computer-Based Education — Rush University ........ 328
*Lisa P. Brenner*

Microcomputerized Tachistoscope to Improve Skills in Algebra ........ 329
*James B. Caballero*

BARNY: A Computer Who Cares About Adolescents' Health ........ 330
*Betty Chewning, Dave Gustafson, Kris Bosworth, Trisha Day, Robert Hawkins*

Use of Video Tape to Present an Introduction to Computing ........ 331
*Alex Rees Clark*

Bi-directional Communications Network Development ........ 332
*Bruce A. Conroe*

Low-cost Pupil Reporting System for Schools ........ 333
*Claudia Devaux*

Introducing Commputer-Based Education — PLATO — at the University of Antwerp (Belgium) ... 334
*W. Dyck, F. Dams*

Computer-Assisted Instruction and Computer-Managed Instruction Project ........ 335
*Margaret "Kit" Ellis*

A NICE Project — A Case Study in Innovation ........ 336
*Analee Elman*

Graphical Explorations ........ 339
*Robert Evans*

Credit Agricole ........ 340
*J. Favry*

Hand-Held Computer (HHC) Training Applications ........ 341
*Frank Giunti, Donald A. Kimberlin*

Computer-Based Instruction (CBI) For the Tow Field ........ 342
*Frank Giunti, Donald A. Kimberlin*

Distributed Instructional System ........ 343
*Frank Giunti, Donald A. Kimberlin*

Videodics Interpersonal Stress Training Assessment ........ 344
*Frank Giunti, Donald A. Kimberlin*

Videodisc Interpersonal Stress Training Assessment ........ 344
*Frank Giunti, Donald A. Kimberlin*

Equipment Independent Training System ........ 345
*Frank Giunti, Donald A. Kimberlin*

Applications of Modern Technology to Army Training ........ 346
*Frank Giunti, Donald A. Kimberlin*

Progress to Date on the Implementation of the Integrated Cybernetic Music System (ICMS) ....... 347
*Sherwin Gooch*

Computer-Assisted Instruction Project ........ 348
*Susan J. Grobe, John R. Tarp*

Microcomputers in Music-Theory CAI: A First Retrospective ........ 349
*Dorothy Gross*

A Real-Time Fundamental Pitch Detector for Use in Computer-Based Music Learning Systems .... 350
*Fred T. Hofstetter*

Tools — Identification and Use in Industrial Arts Education ........ 351
*J. F. D. Ilott*

Evaluation of a Spatial Data Management System for Basic Skills Education .................... 352
*Greg P. Kearsley*

Cross-Lingual Mathematics Information System .......................................... 353
*Jim Kelly*

Field Test of a Microcomputer-Based System for Computer-Assisted Instruction in Music .......... 354
*Wolf Kuhn, P. Lorton*

Mathematics in Biology: Computer-Controlled Videodisc Materials for Women and Minorities .... 355
*James M. Laffey*

The PLATO Calculus Course ................................................................ 356
*Max Larsen*

CAI House — Renting vs. Owning — College Level Housing ............................... 358
*Mildred C. Luckhardt*

CAMI at Johnson County Community College .............................................. 359
*Wayne Mackey, Sue Lindsay, Harold Harp*

CAI at Los Alamos National Laboratory .................................................. 360
*Mary S. Manteuffel*

Teaching Descriptive Inorganic Chemistry in First Year University Using High Resolution Graphics ........................................................................... 361
*John S. Martin*

CAI Tutorials in Introductory Chemistry ................................................ 362
*John S. Martin*

Robust Courseware Development ........................................................... 363
*Paul G. Matthews*

CAI at the UND School of Medicine ...................................................... 364
*Mary Fredricks Medal*

Melodic Dictation Via PLATO: A Progress Report ....................................... 365
*Steven R. Newcomb*

Computer-Based Instruction Project (CBIP) .............................................. 366
*Thomas C. Reeves*

French Unstable e ........................................................................ 367
*Bernard Rochet, Jim Horne*

The Use of Computer-Aided Instruction in the Nuclear Industry ........................ 368
*D. L. Rogosky*

Using PLATO to Teach Library Research Skills .......................................... 369
*Judith Sandler, Patricia Arnett, Patricia Fitzgerald*

Delivery of Continuing Medical Education by Computer-Assisted Instruction .................. 370
*Ernest N. Skakun, Milt W. Petruk, Colin Park, R. S. McLaughlin*

Development of Microcomputer-Based Instructional Systems ................................ 371
*Nick Van Matre*

Charon: A Curriculum Management System Based on Instructional Groups ..................... 372
*Richard L. Venezky, Gary A. Feurer, Kenneth Kahn*

Sight-singing via PLATO A Progress Report ............................................. 373
*Bradley K. Weage*

Republic VII: A Portable Music Station for Use With a Remote Host Computer Library .......... 374
*Bradley K. Weage*

# INDEX OF AUTHORS

| AUTHOR | PAGE |
|---|---|
| Abernathy, David R. | 52 |
| Arenson, Michael | 228 |
| Bagnall, K.M. | 132 |
| Belanger, Jean | 37 |
| Bonar, Jeffrey | 185 |
| Boody, Charles | 97 |
| Bork, Alfred | 213 |
| Brandt, Richard C. | 193 |
| Brebner, Ann | 259 |
| Bregar William | 113 |
| Brooks, Sarah | 212 |
| Bruckhart, Glenn | 93 |
| Buck, John A. | 34 |
| Burge, Mary Jane | 109 |
| Chapin, Thomas | 174 |
| Christensen, Mark J. | 210 |
| Conklin, D.N. | 136 |
| Cook, James | 263 |
| Creutz, Alan | 89 |
| Crown, E.M. | 161 |
| Dallman, Brian | 62 |
| Davis, Richard W. | 174 |
| Diemer, Jans | 1 |
| Dimas, Chris | 311 |
| Doherty, Maryanne | 161 |
| Dumper, Chris | 122 |
| Eaton, John | 235 |
| Elman, Annalee | 270 |
| Evans, Richard | 298 |
| Farley, Arthur M. | 113 |
| Fehling, Michael R. | 283 |
| Fisher, Francis D. | 16 |
| Fitting, Marjorie A. | 218 |
| Francis, Larry | 47 |
| Gaddis, Karen | 85 |
| Gerhold, George | 12 |
| Green, John | 42 |
| Gress, Eileen | 179 |
| Hair, Harriet I. | 222 |
| Hallworth, H.J. | 259 |
| Hannaford, Alonzo E. | 22 |
| Hannah, K.J. | 136 |
| Heimer, Ralph T. | 231 |
| Holmes, Glynn | 245,252 |
| Holmes, Zoe Ann | 170 |
| Hopper, Judith A. | 93 |
| Hudson, W.J. | 57 |
| Jenkins, Tracie | 126 |
| Johnson, William B. | 85 |
| Johnson, Jerry | 219 |
| Jongejan, Tony | 219 |
| Kadner, Kathleen | 148 |
| Kallal, A. | 132 |
| Kearsley, Greg | 25 |
| Kheriaty, Larry | 12 |
| Kidd, Marilyn | 245 |
| Klotz, Eugene | 220 |
| Lantz, Brian | 113 |
| Larson, Donna E. | 141 |
| Lemke, Cheryl | 219 |
| Lorton, Paul | 235 |
| Mayhew, Frances Whitaker | 166 |
| McLaughlin, Donald H. | 276 |
| Montgomery, Anne | 74 |
| Mullinix, Patricia M. | 231 |
| Munro, Allen | 283 |
| Mydlarksi, Donna | 252,259 |
| Pagliaro, Louis A. | 151 |
| Paramska, Donna | 252 |
| Pieper, William | 70 |
| Richards, Griff | 37 |
| Richardson, J. Jeffrey | 65 |
| Rollingher, I. | 132 |
| Rubin, Eric | 185 |
| Russell, William E. | 104 |
| Salt, Norman F. | 239 |
| Schaeffer, Karen E. | 292 |
| Seebach, J. Arthur | 203 |
| Shearman, Gareth | 267 |
| Shirer, Donald L. | 5 |
| Smith, David A. | 217 |
| Soloway, Elliot | 185 |
| Spannaus, Timothy | 306 |
| Squire, Enid | 42 |
| Steffen, Dale | 79 |
| Szabo, M. | 132 |
| Taber, Florence M. | 22 |
| Terry, Kathleen A. | 52 |
| Towne, Douglas | 285 |
| Trollip, Stanley R. | 85 |
| Wenger, Ronald H. | 197 |
| Wheeler, Donald A. | 109 |
| Woodson, M.I. Chas. E. | 289 |
| Woolf, Beverly | 185 |
| Zimmerman, Michael J. | 31 |

# AUT VINCERE AUT MORI

JANS DIEMER

CAMOSUN COLLEGE

## ABSTRACT

Private industry is rapidly customizing communications technology to educational needs at hardware and software levels. The potential loss of their mandates by educational institutions can only be avoided if educators are more open to change and become more directly involved in the continual development of new courseware. The learning environment requires alterations and facilitators must assume roles of responsibility in the emerging information society.

Hardware-independence of courseware should be maximized if the impacts of economical forces on the ability to deliver are to be minimized.

The K-12 process is presenting post-secondary institutions with a new breed of candidates. Applicants will increasingly be looking for skills that are immediately marketable.

Theoretical studies will be demanded by proportionally fewer students. Software evaluation, maintenance and customization will require a new type of graduate, who can communicate effectively, understand people and organizational problems, and is secure in daring to offer solutions of unquestionable quality.

This paper summarizes relevant changes in our professional/technical environment and includes some basic, global recommendations for coping.

I wish that in the next few minutes I could be everything to all present, and I may get close to success. Those who want to be critics will have ample substance. A few will go to sleep. Some will count my distracting habits. Some of you will want to do like me: we'd like to leave the impression that we know it all. Our being here however speaks otherwise. Some of us are scared. One day we hear that PLATO is the ultimate, the next day we are told that it might cost too much. A bit later we see a fantastic demonstration, and even get to touch the screen. Within a few moments the telephone system linking us to the host dies, and we touch a cold impersonal dark screen. If we don't learn about new things, investigate, dabble a little and learn the CBL vernacular, we might just be doomed. If we do, those even more scared among us or who control the purse strings might damn us. That's the scenario for all of us who want to be literate, or just want to do a better job if we only knew how to.

When we read a lot, because we passed the literate stage and were mesmerized by new horizons, we have new reasons to be afraid. Scarcely have we read a very convincing argument for a new software development, or for a new protocol for communication, when yet another reputable informer tells us that so and so has invented this neat shortcut, deflating the convincing argument. Today ADA is alive. Tomorrow 'she' may be dead. Will we ever become up to date? What in the world is Ethernet?

It's not that long from von Neumann to a 32-bit micro. Within the timespan of the last five years we have witnessed 99% of all changes in computer technology, and I heard someone say at a meeting around New Year 1982 that 'of all computers, big and small, in existence today 90% were not born a year ago'. I'll question that with you, but there are inherent messages. On several occasions time-management experts have suggested that many of those who program computers no longer have a chance. By the time they are well underway on an assignment something changes uncontrollably fast. Often it is an improvement in methodology or in hardware, in part the fruits of their own handiwork, that becomes their undoing.

A major difficulty for us all is the rapid technological change that is taking place in the home and in the K-12 facility. I was teaching an introductory programming course a little over a year ago, and I asked my students to suggest how we might achieve our learning objectives. In this small class of 32 students 11 had a microcomputer. Our course became for most a 'do it at home' unit. The need for a special terminal-room reservation disappeared, and so did the value of my practical course notes which were somewhat governed by our VAX-systems. Obviously my notes would have to become more or less hardware independent. I was very pleased to see that most of the students produced work at a level of sophistication I had thought to be near impossible to achieve in 14 weeks. Microcomputer retail businessmen indicated that the number of systems going into homes in B.C. during the last Christmas season was quadruple that number placed during the entire year 1980. Remember that this new base of personal technological sophistication began to form in 1971 when Intel commercialized its first microcomputer on a chip, the 4-bit 4004. Now just over a decade later, we have gone through the 8- and 16-bit machines, are frequently reading about new maxi-micros with a 32-bit base, 128K or more of memory and many megs of disk-based storage, and find that user societies start up as soon as a new system hits the market.

Educators at all levels will feel the change currently effected in expertise and expectation of students currently passing through the K-12 system. Minnesota, British Columbia and Alberta, to name a few self-determining educational entities have facilitated access by youngsters to computing through hands-on experience with microcomputers. Graduates of this system enter post-secondary institutions with expectations of advanced studies in computing. The responsibility of Colleges and Universities is now broadened: post-secondary computer education will soon need to start at a higher level than before for most highschool graduates. For a few, unlearning of poor programming habits must first occur. This implies the need for intensive articulation, standardization at all levels, for the education of those who teach computer science in the schools and for harnessing and reorientating of the entrepreneurial energies of pedagogues who suddenly have a new tool which is widely tauted as unavoidable. It has been said that the 11-year old in 1982 likely knows more about, and is more familiar with, computers than the 19 year old who left school in 1981. One teacher confided that he learned more from his students than vice versa.

I do not wish to belabour the microcomputer at the expense of other influential events. Networking between information bases, and between hardware systems that are integral parts of data caches, has greatly increased the potential availability of knowledge and information at any location were telephone, television or satellite signals can be received. It will not be long before the insatiable hunger for new technology fosters the marketing of reasonably priced 2-way audio/video devices. We've already seen the possibilities of micros talking to mainframes, talking to other micros, controlling videodisks, supporting videotex graphics, etc. It can be done, but are we doing it? When we apply new technology, is it of high quality? Is it used as a means, rather than becoming an end?

I ask these questions legitimately because of the discrepancies that continue to exist between the promises on the one hand, and implementation intensities on the other, of auto-tutorial, Keller, computer assisted, ETV, and otherwise modified instructional and learning strategies. In a recent conversation with potential users of humanties and science courseware aimed at the post-secondary level that was available off some software-house shelf, it became clear that one of the reasons for limited application of canned courseware, whether computer-based or supported by other technologies, may well be the traditional resentment by many educators to use anything that is not of their own creation. In addition, the incompatibility of many videorecorders and players that existed early in the life of educational television has not been forgotten. There are many skeptics, and as a result many traditional educational practises remain entrenched, inviolate so to speak to 'inappropriate' influences and changes.

But can we afford to sit back? Should we contribute our share to the next generation's formative years and beyond or watch the world go by ever more rapidly? It has been widely held that recent relaxation in communication related legislation will increase certain monopolies, but will also speed up the entry of new communication devices into the home. The physical profile of educational institutions may consequently change. The learner will become quite comfortable with the new technology that permits learning at home. There will be economies/political pressures on Colleges and Universities to provide the maximum quality educational service for minimal costs. I assume that all institutions are as efficient as mine, and therefore less costs can only result from either a reduction in services to students, or from a much greater contribution to the costs by the user. Wherever new technology can help it will likely succeed.

One major requirement of the educator is the review of learning theory, and, as Mitzel observed at the 1981 Conference in Atlanta, the development of a 'strong and viable theory of instruction'. The learner of the 80's will insist on individualization of courses of study, timetables, strategies for learning, and will, I predict, want to fully utilize personal resources that were marketed and purchased as educationally valid. In essence, then, the educator will be faced with entirely new influences most of which have not been analyzed systematically in the literature.

What can be done today to provide for a better learning-result tomorrow? The answer to that question requires a definition of 'better learning-result".

Better, being comparative in nature, refers here to more efficient, more effective, more relevant, more career oriented, more pedagogically sound, or whatever else you wish to substitute, but also including connotations of cost to the user.

Learning-result is usually the outcome as seen by the facilitating institution and its governance bodies, and is often measured in terms of immediate usefulness. Lately usefulness is a synonym for employability. Now we can attempt to answer the question. We can immediately begin to convince ourselves that as sincere educators we cannot allow private industry to encroach on our courseware mandate, unless we agree that education does not lie at the roots of a free and prosperous society. We can begin to establish a behavioral consensus that will force some of us to pull up stakes, or what have you, but aimed at education for the 80's and 90's. Include if you will greater flexibility on the part of the educator and of the institutions more concern for the learner and less for turf, and an awareness that, but for a chosen few, specialization in technology may lead to redundancy and obsolescence.

Out of this flows the suggestion that those who are being trained to educate not become computer programmers, but become learning and teaching and guiding and counselling experts who use new technology to the fullest. I see the need for software and courseware analysts concerned with the application parameters, rather than with benchmarks. It is clear that technological innovations will reduce the requirement for traditional programmers and for in-house development. Software maintenance will be required more and more as software houses become more sensitive to educational needs.

Another suggestion is that we limit but not totally remove our use of jargon in order that the danger of familiality, which can make a monster out of automation, is minimized. That is to say, our documentation should be user-friendly, but not so mushy (Ko et al, 1982) that it makes the system completely transparent, and becomes an unreasonable burden on machine resources.

Another suggestion is that we ensure at least a literacy in new technology in all those who make related decisions. If we don't, legislation, policy making, or majority decision making at any level could result in debilitating restrictions on and most unfortunate delays in, implementation of new technology at a time when at least the educational process and learners can benefit greatly.

Another major point bears mentioning here. Education, in part an information technology itself, is the only unbiased route by which the general public can become informed about the drift towards the information society. Thus education is presented with a leadership role that cannot be conceded to the political or private socio/economical arenas. Educators can show the past and present, and project on that basis into the future; other agencies may well have to suppress knowledge for the sake of survival.

Finally I would recommend that you read even more than you have up to now. If we wish to retain our position as enablers of intellectual and skill development then we must be cognizant of all forces at work around us. I have personally taken great delight in the collective inputs of several micro-related journals, editorial comments in many technical periodicals, the writings of Jean-Jacques Servan-Schreiber, C.P. Snow, and T.H.E. Journal, Educational Technology, CAUSE/Effect, JCBE, Computerworld, and proceedings of as many appropriate conferences as I can get my hands on. I am beginning to get an idea where we might be by 1986. Hence my concerns shared with you today.

Those among us who have had wide and in-depth experience with applications of new technologies to learning and/or instruction, whether at the practical or theoretical level, will have to be aware of the apprehension new technology can easily create for the uninitiated. In the foregoing paragraphs a potential role for practitioners involved in the education of educators was alluded to. This role is, in my opinion, more to share the successes of applications and the psychology of living with them, than to explain at profoundly theoretico/technical levels what brought success about. In addition we should, as educators, ensure the learner in the 80's and 90's that a life-long learning skill is being passed on as well as the contents of the current course of studies. The economic stresses we experience at all levels will require this bonus. We are speaking about a very important guarantee: if holistic learning and growth cannot be provided in the traditionally funded and designed manner, then at least self-initiated further growing should be enabled if at all possible.

It is with the assistance of new approaches to instruction and new appropriate applications of Communications Technology that the facilitating role of the professional educator can be realistic. It can not be so without the humaneness notably absent in machines.

If this short collection of thoughts has touched you in only one aspect of your very being, personal, professional or otherwise then I shall be grateful. If not, then appreciate the fact that I had to get this off my thorax in a controlled manner.

I am leaving for you to take away, a Summary of the Salient points and recommendations, in appendix form. For the eager novice a short reading list is also available. These materials are available upon request in writing.

# EVALUATION OF COMPUTER-BASED INSTRUCTION SYSTEMS

Donald L. Shirer

Dept. of Electrical Engineering
University of Arizona, Tucson AZ 85721 (602-626-3302)

## ABSTRACT

More than 20 computer-based instruction systems were evaluated as part of the selection process for a large lesson development project. Their capabilities differed widely and were not always correlated with cost per user-hour. A description of the evaluation process is given, including cost determination, so that the evaluations may be adapted to local needs.

***

Although computers have been used to aid the educational process since the late 1950's, only recently have enough examples of computer systems suitable for delivery of computer-based instruction (CBI) appeared on the market to give the educator a meaningful choice of delivery media. Functional descriptions of computer systems can be very misleading when judging their suitability for CBI, which depends on a proper match between hardware and software. The study reported here was undertaken to select a CBI system for a very large lesson-development project, and actual lesson material was prepared to test most of the systems mentioned. Although our own evaluations will be presented at the session, the main thrust of this paper is to describe the evaluation methods so that other sites may select systems suited to their local needs. A printed summary of our evaluations is available upon request. It is hoped that this will be helpful to novices in the computer education field, as well as to computer professionals less acquainted with the demands of educational use.

First, it is necessary to define the special needs of Computer-Based Instruction. While it is possible to use almost any computer to create simple drill-and-practice routines, or to let students create their own programs (both worthwhile activities), CBI is generally recognized to go far beyond such simple tasks. Common uses[1] include interactive simulations, tutorial "dialogs" between student and machine, on-line testing and the managing of instructional resources, as well as facilities for instructiors to supervise the progress of their classes.

## DISPLAYS

The most visible indications of computer system performance, and the factors which most affect lesson design, occur at the user-machine interface. Nowdays, most computers present data to their users via an electronic display. Educational needs often demand complex displays containing graphs and figures, for which the low-resolution of many hobby computers[2] may be inadequate. The table below suggests a conversion of quoted screen resolution in both alphanumeric lines and columns and in graphic pixels or dots to an ABCDE scale more familiar to teachers. E is inadequate, D is barely adequate and C is deemed satisfactory for average educational use. Resolutions rated B or A are preferable for fine graphic or pictorial displays.

| Lines | x Columns | Rating |
|---|---|---|
| 16 | 24 | D |
| 20 | 40 | C |
| 32 | 64 | B |
| 32 | 80 | A |

(This does not imply that you should fill 16 lines with 24 characters in any display, much more white space is always preferable. However, this much resolution is needed to see fine detail in pictorial displays.)

| Horizontal | x Vertical | Dots | Rating |
|---|---|---|---|
| 256 | 128 | | D |
| 320 | 256 | | C |
| 512 | 340 | | B |
| 512 | 512 | | A |

The ability to substitute user-designed character sets for languages other than English, for mathematical or scientific symbols, or for creating lifelike pictures is essential for many CBI uses. A rating for "charsets" is included in the sample evaluation list shown in the appendix. Generally a "B" for available and a "D" for not available is sufficient.

Selective erase, the ability to erase a selected portion of the screen without affecting the rest, is essential for CBI use. Luckily it is available on most raster displays, but storage tube displays without selective erase must receive an E in this category. The width of the communication channel between computer and screen can be estimated by measuring how rapidly the computer can fill an entire screen with alphanumeric characters.

| Rate | Characters/sec | Rating |
|---|---|---|
| 110 baud | 10 | E |
| 300 | 30 | D |
| 1200 | 120 | C |
| 2400 | 240 | B |
| 4800+ | 480+ | A |

Ordinary telephone lines are generally limited to 1200 baud, adequate for CBI use, but computers closely coupled to their displays can produce much better animations.

Color is a controversial issue. It is eye-catching, but easily overdone. Underlining, special fonts, boxes or blinking can be used as emphasis in monochrome displays. If you need color or special output features like an audio tone generator, include a rating line for this on your evaluation list.

USER INPUT

The use of a typewriter-like keyboard is standardized in the computer industry, but unfortunately their layout and quality are not. Look to see where the 0,1,+,- keys are located. Is the ENTER(or NEXT or RETURN) key conveniently placed? Downgrade keyboards with mechanical contacts which will sooner or later prove troublesome. Does it "feel" right? Additional input from a touch panel, graphic tablet or "game paddle" to enable the user to "point" to specific objects on the display is especially valuable for young users or those who cannot (or will not!) type. Such extra input is so generally useful that you should leave an explanatory line for it on your evaluation list.

HARDWARE

Computer experts spend much time evaluating the electronic and mechanical aspects of a machine, but estimates of "computer power" are usually not very meaningful in CBI use because many important facilities are furnished in software. As long as typed keys are echoed on the display in no more than 0.2 sec, and question responses are evaluated in one second or so, CBI users appear to be satisfied. Even systems using 8-bit microprocessors can accomplish this if programmed intelligently, so all systems mentioned achieve at least a C rating. Large time-sharing machines can furnish bursts of high power when needed for answer analysis, and microcomputers with 16 bit processors, or several 8-bit chips deserve a higher rating.

One hardware aspect difficult for even experts to predict is the quality of the mechanical or electrical construction. Mechanical keyboards and cost-cutting disk drives cannot be expected to stand up to the constant use (and abuse) generated in an educational environment. Many hobby computers had to be rated D-C in this area, but lately quality has been improving for models intended for business use. Hall-effect or capacitive keyboards and cushioned head disk drives used in professional equipment can be expected to require less maintenance and give longer service life, and therefore would contribute to A-B ratings in this area.

When evaluating CBI systems, if you have special requirements for driving or accepting data from peripheral devices like slide projectors, videodisk or tape units, music synthesizers, etc, be sure that the appropriate connections and software exist. Interfaces for such devices are either available or can be easily constructed for all computers evaluated.

ANSWER ANALYSIS

A "system" intended for CBI use can hardly be called that if it can only test for exact matches with numbers or character strings. This is why most standard computer languages like BASIC or FORTRAN are not well suited for writing instructional lessons. While examples of good CBI lessons using these languages exist, it costs perhaps ten times the effort and time to create a lesson compared to that needed on a system with well-designed software. A suggested rating scheme for answer analysis facilities is listed on the next page. Each line includes by implication the properties listed in previous lines.

| Answer Analysis Capability | Rating |
|---|---|
| Exact number, character string match. | D |
| Word and phrase match. Spelling checks. Numbers+tolerances. | C |
| Concepts, synonym lists, algebraic formulas and expressions. Order checks. | B |
| Numeric or Algebraic values with units. | A |

The ability to examine and modify the user response is essential if proper answer analysis facilities are not provided by the system. Most normal computer languages have a D rating in this area, unless many hundreds of man-hours are spent in writing general subroutines to do the answer analysis yourself. Special CBI languages do better. The PILOT language would rate a C- here, and the TUTOR language used with the PLATO system would have an A rating for answer analysis.

CALCULATION FACILITIES

Most standard programming languages rate at least a B in their provision for calculations by the author, and a D in the amount of calculating which can be done by a user when entering his response. Most versions of PILOT or early Coursewriter-like languages also rate D's in both categories. TUTOR would rate A/B.

| Calculation Facility | Rating |
|---|---|
| Integer Arithmetic | E |
| Real number (6 digit) | D |
| Expressions (6+3/5) | C |
| Symbolic (a+b/2) | B |
| Matrix + other facilities | A |

GRAPHICS

Systems differ in the provision of graphic design ability for the author. Ease of graphics use is essential for good interactive lesson design. An estimate of the needs for a general purpose CBI system is shown below. If you need a system for primarily scientific or mathematical work, use the ratings in the last column.

| Graphic Capability | Normal | Scientific |
|---|---|---|
| Lines, characters only. | D | E |
| Lines, circles, boxes, character sets. | C | D |
| Rotated or relative graphics. | B | C |
| Graph making and animation facilities. | A | B-A |

EASE OF LESSON PRODUCTION

This feature of a system can only be judged accurately by actually writing CBI material on the system in question. A sample unit involving both text and graphic material and several types of answer analysis was coded for as many systems as possible, in many cases using a programmer experienced on that system as a guide. A rating was based on the ease of programming the lesson unit and coding it onto the system. This rating was influenced as much by the software operating system as by the actual language used for programming. Programming time ran as short as 2 minutes (on the PLATO system) to 30 minutes on some especially "unfriendly" systems demanding a switch between several different editors and linkers. A high rating here represents the least time, and will be reflected in lower lesson design and coding costs during CBI production. Many systems offer simplistic design systems for beginners (EASY, TACH, UAS, CREATE, etc) touted as "easy-to-learn" and "simple-to-use". In every case, it was judged that these so-called "easy" systems were actually more difficult to learn than a suitable starter subset of a CBI language. They also restricted a programmer to a format suitable for only a small fraction of CBI lesson material, and involved an extra-operating system layer which slowed production of lesson material. Luckily only one system (UAS) forces the author to use such a "template" system, so most authors would do well to ignore the "easy" systems and use the more powerful facilities available. A separate rating should be provided for the editor and operating system facilities, since they are most often used by authors. The standard CP/M system used with hobby computers would only rate a C, while the powerful file and operating systems provided on some time-sharing systems would be in the B-A range. (An A-rated system would not involve the user with a job-control language!)

STUDENT FEATURES

Turning now to features which make it easy for a student to use a CBI system, by far the most important is the "friendliness" or ease of use of the operating system. A student must identify himself to the system, pass an optional security check, and then be sent to a lesson of his or his instructor's choice. An operating system which requires a student to know many details of the computer itself must be rated E. A system which provides these facilities without burdening the student with unnecessary detail would be rated C, and one which permits the student to sign in with his own name, choose his own password, and not need to be taught anything about the computer itself would rate a better evaluation. For response entry, students should be able to correct mistakes easily, type standard math symbols like pi which are not on the keyboard, and standard diacritical language marks for a C rating.

The student should also be able to leave a lesson partially completed, retaining a part-score or status, so that he may complete the lesson at a later date. The operating system should provide such automatic record keeping facilities(B rating) or allow the author to provide them (C-D). Finally, there should be a limited amount of student-controlled branching, so that a student can press a function key (or type HELP) and receive a hint. Similar entries could initiate a review sequence, see a data table, or provide a calculator. Oddly enough, even some otherwise sophisticated CBI languages do not provide this feature, and must be rated D-E here. The TICCET[3] system originally inisisted that progress through a lesson be entirely student-directed, a philosophy with which many instructors differ. A more traditional language system is now being developed for the TICCET system, which permits the author to choose his own lesson structure.

AUTHOR FEATURES

Separate ratings should be included in an evaluation list for the system editor and for testing/debugging features. Programmers interact with the editor more than any other program, and a good screen oriented editor (rating B-A) should be rated higher than a line-by-line editor(C) because it increases productivity. The ability to store student responses during the testing/debugging stage which accompanies any lesson writing project is necessary for proper lesson evaluation. If student responses can be stored and recovered, rate this feature C. If they can be separated into anticipated/unanticipated, correct/incorrect, help requests, logons, logoffs, etc., the rating would be B-A. If you do not intend to write your own lessons, you may weight this feature lower, but do not omit it entirely.

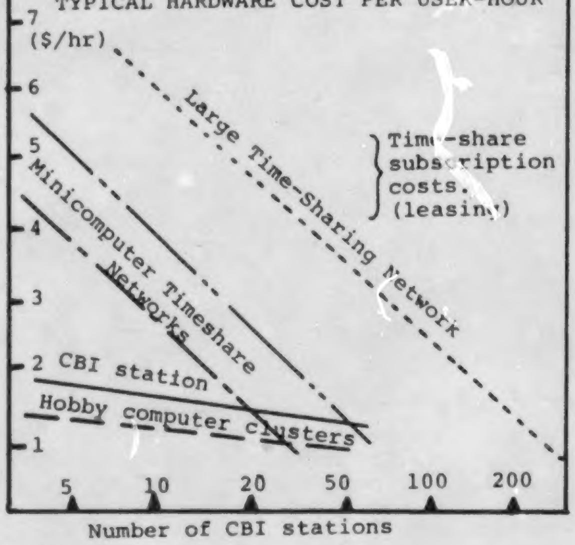
INSTRUCTOR FEATURES

The ability to select a course or sequence of lessons for a class or individual student, or for the smaller systems, the ability to merge lessons onto a disk and select them, must be rated by actual trial. Many systems provide no help to the instructor in this area, and the teacher must write her own routing program. These would be rated D or C. If automatic selection by class is provided, give a B rating, if individual sequencing is possible for each student, rate this an A.
If you are going to use lessons written by others, possibly the most important line of your evaluation should be the length and quality of the existing course catalog in your area of interest. The PLATO system has over 10,000 hours of lessons on tap, but some newer systems have not yet provided for a users group to exchange lessons among their customers. Remember, not all published lessons are of high quality! Finally, a line should be provided for evaluation of the on-line record-keeping available to the instructor. Rate E if there is none, D if only time on is available, C if the instructor can ascertain which lessons his students have completed and their scores. Automatic averaging of scores for the class and "gradebook" facilities would move the rating into the B-A region. Another rating line should be provided for registration facilities. It should be possible to register students into classes or groups easily, change their records if necessary, and optionally send them messages.

COSTS

Costs can best be compared between dissimilar systems by first calculating the total costs of all services for the lifetime of the equipment, and then dividing the average cost per year by the number of user-hours provided by that equipment during a single year. Prices change so rapidly in this field that you must contact a manufacturer's representative for up-to-date figures. Typical figures for the hardware costs alone of several prototype systems are shown in Figure 1, based on terminals being available 40 hours per week, 45 weeks per year.

Figure 1
TYPICAL HARDWARE COST PER USER-HOUR



Hardware costs should include maintenance or service contracts and site preparation. The costs of operators and attendants for time-share systems should also be included here. Under these conditions, hobby computers can average $1 per user-hour or less to operate. The dashed line in Fig.1 includes a central file controller to tie these into a local network. Microcomputers designed especially for CBI systems average under 2$ per hour to operate, even when tied into clusters. This is less than the cost of an average class-hour of instruction in a high school! Two prototype minicomputer networks required from 20-50 terminals before they became cost-competitive with stand-alone stations. A time-sharing network based on a large mainframe computer for which subscriptions are sold at $3 to $5 per hour takes more than several hundred terminals to bring its costs down lower than the mini nets. If you are located at a distance from the central computer, you may also have to include communication costs which can go up to $2000 per year per terminal.

To these figures, you must add the cost of CBI software and lessons, classroom monitors, disks, paper, etc. Lesson royalties may be typically 25 cents per user-hour when using published lessons, much higher if you write your own material and do not spread the cost by publishing it. A typical figure is $1000-2000 per lesson. See Appendic C for an example.

CONCLUSIONS

The complete evaluation list used in this survey is shown in Appendix A, and the systems evaluated are listed in Appendix B. Since you should weight your evaluations according to your own needs, I will quote only a few general conclusions which would seem to be valid for all users.

CBI NETWORKS

The first three networks listed in Appendix B rated highest in all categories (student,author,instructor) except cost. Of these, only the PLATO[4] services are available by subscription. These would seem to be the systems of choice if large lesson-development projects were planned. Of the systems based on IBM machines, the PHOENIX[5] system has better facilities than the others, and is worth considering if you have a 360/370 with unused capacity.

STANDALONE CBI STATIONS

Of the standalone stations specially designed for CBI use, the REGENCY[6] provides both author and lesson delivery services as good or better than that of the PLATO system at a reasonable cost. The MICROPLATO[7] stations provide good lesson delivery from diskettes, but must be connected to a PLATO system to author new material, and so have a higher overall cost. Both systems lack some essential instructor options, which will hopefully be provided in the future by forthcoming cluster controllers for each system.[8]

PERSONAL PROFESSIONAL COMPUTERS

There are many powerful computers in this group, but most are eliminated on price or for lacking CBI software. The TERAK[9] machine has reasonable hardware and graphics facilities (though they are overshadowed by recent introductions) but the only CBI software available is a primitive template system providing far fewer facilities than the previous group.

HOBBY COMPUTERS

The ubiquity of these machines would seem to make them good candidates for CBI, and it has been proven that with enough care, it is possible to do CBI on microcomputers![10] However, most hobby computers lack good graphics facilities as well as the author/instructor features provided by the CBI machines. A version of the simple PILOT CBI language is available for most of the computers listed, and would certainly seem preferable to trying to program CBI material in BASIC, the one language shared by all machines in this group. However, if you write your own lessons, it may take from 5 to 10 times the time and effort to program in PILOT or BASIC compared to a good CBI language. The Bell and Howell system is the closest to providing a complete student/author/instructor interactive facility, although much better systems are on the drawing boards of several manufacturers.

Many publishers are bringing out classroom "systems" using these machines for K-6th grade mathematics and English, spreading the higher cost among many customers. It is not the purpose of this paper to evaluate these courses, but it is worth noting that salesmen for each one contacted refused or were unable to supply proof that they actually taught what was advertised, or even the average length of time a student would spend taking a lesson.

TRANSPORTABILITY

Unfortunately at this time it is not possible to guarantee that a lesson written on one computer can be transferred to another and continue to work, even in PILOT or BASIC. This problem must be addressed before CBI becomes widespread. To transport a program to a system using another dialect of the same language type, a good programmer might have to spend from 10% to 20% of the time originally put into creating a program to revise it. Transportation is easier among the PILOT group, the BASIC group or the PLATO-like group of languages than between these groups. For this project, the final decision was to develop the lessons originally on the PLATO system, because of its fine author/instructor/testing facilities, and then to transport them to standalone CBI stations with similar languages which offer lower cost lesson delivery.[11]

## APPENDIX A: Suggested Evaluation Form

**Display** Rating
Resolution (line/col) ____
Resolution (graphic) ____
Display Rate ____
Color (optional) ____

**User Input**
Keyboard Layout and "Feel" ____
Extra Input (type______________) ____
Touch Panel ____

**Hardware**
Keyboard Construction ____
Disk Drive Quality (if disk) ____
Estimate of Reliability ____
Computational "Power" ____
Special I/O Interfaces(opt) ____

**Student Features**
Answer Analysis Features ____
Calculation Facilities ____
Graphics Capabilities ____
Special Characters(charsets) ____
Signon Features ____
Lesson Completion and Restart ____
Student Directed Branching ____
Operating System "Friendliness" ____

**Author Features**
Operating System Ease of use ____
File System Power and Ease ____
Editor Features ____
Author Language Ease of use ____
Author Language Facilities ____
Author Calculation Facilities ____
Compiler Speed (if used) ____

**Instructor Features**
Student Records(time/score/etc) ____
Student/Class Identification ____
Automatic Record Statistics ____
Ease of Student Registration ____
Ease of Lesson Selection ____
Student Response Collection ____
Testing/Debugging Facilities ____
Message Services ____
File Forwarding and Storage ____
Size of Lesson Base ____
Quality of Lesson Base ____
Existence of User Group ____

**Cost**
Hardware $/user-hr____ Rating__
Lessons $/user-hr____ __
Supervision $/user-hr____ __
Total ____ __
Service (availability/quality) ____

## APPENDIX B: Computer Systems Evaluated For CBI Use

**Computer Timesharing Networks**[12]

PLATO. Distributed commercially by CDC Education Systems,HQA02Q, PO Box O, Minneapolis MN 55440.

SIMPLER. Modular Computer Systems, PO Box 6099, 1650 W. McNab Rd., Ft. Lauderdale FL 33310.

DIMENSION. Digital Equipment Corporation, 129 Parker St., Maynard, MA 01754.

TICCIT. Sold commercially by Hazeltine Corp. 7680 Old Springfield Rd,McLean,VA.

MCDONNEL-DOUGLAS Corp., 700 Broadway, Suite 617, Denver CO 80203.

BOEING Computer Services, Box 24346, M59A-90, Seattle WA 98124.

PHOENIX. Goal Systems Corp., 1865 Fountain Square, POBox 29481,Columbus OH, 43329.

CAN-8. Distributed either by Honeywell Computers or by HOMECOM Learning Systems Ltd., 91 Glencameron Rd, Thornhill, Ontario, Canada L3T-1N8.

**Standalone Microcomputers Especially Designed for Computer-Based Instruction**

MICRO-PLATO. Control Data Corporation. See address under PLATO above.

REGENCY. Regency Systems, 1610 Interstate Drive, POBox 3590, Champaign IL 61820.

**Personal Professional Computers**

TERAK 8510. Terak Corp, 14405 N. Scottsdale Rd., Scottsdale AZ 85260.

HP-85. Hewlett-Packard Co. Dept.287E, Corvallis, OR 97330. (The newer HP-87 was not evaluated. The HP 9826 and 9836 provide better facilities at higher cost)

TEKTRONIX, WICAT, APOLLO, THREE RIVERS, ZILOG, XEROX and many other manufacturers also produce instruments in this category, but they have no provision for educational CBI use.

**Hobby Computers**

APPLE II (and III). Apple Computers, Inc., 10260 Bardley Dr., Cupertino CA 95014.

IBM PC. These are sold through local Computerland and Sears Computer stores. IBM Info.Syst. Division, POBox1328-P, Boca Raton, FL 33432.

RADIO SHACK Education Division, 1300 One Tandy Center, Ft. Worth, TX 76102, makes TRS80-Model II and III and Color computers. The newer 16 bit model was not evaluated.

ATARI 800. Atari, Inc. 1265 Borregas Ave., Dept. C, Sunnyvale CA 94086.

TI 99/4. Texas Instruments, Inc. POBOX 53, Lubbock Texas 79408. This division supports the LOGO language, especially useful in training young students about computer languages.

PET. Commodore Computer Systems Div., 300 Valley Forge Square, 681 Moore Rd, King of Prussia, PA 19406.

PASS. Bell and Howell Corp. 7100 N. McCormick Rd, Chicago IL 60646. This is a software author-lesson delivery system which runs on Apple II computers.

MANY other microcomputers made by smaller manufacturers were considered, but lengthy evaluations were not made since their capabilities were similar to the machines mentioned in this section. This is not meant to slight their product. However smaller manufacturers sometimes have trouble supplying service to their machines. If there is any doubt about the availability of service, add a line to your evaluation sheet about this.

APPENDIX C: Example of Computing Costs.

Suppose a standalone CBI machine costs $8000 with one disk and $9000 with two disks. If it is desired to author lessons, two disks are necessary. Estimating a 7 year service life, the hardware cost is $1300/yr. Maintenance (mostly for the keyboard and disks) might cost $500/yr if you have a local technician service the machines on call, more if you have a manufacturer's service contract. You should add perhaps $100 per year for routine preventive maintenance. Total hardware costs are then $1900/yr/station. We will assume tables and space are available and that no special installation is necessary. If the stations are available 40 hr/wk, 45wk per year, each can provide 1800 user-hours of service. Let us reduce this to 1600hr to account for typical schedule variations. (Even this is conservative, at U.A. our CBI stations are available more than 3800 hr/yr). The cost per user-hour for hardware is then $1900/1600 = $1.20/hr.

The cost of a student assistant to monitor a classroom with 32 stations and to handle scheduling and instructor services is only $0.15/hr/station.

The cost of lesson material might average about $0.25/user-hour if bought or leased from a publishing house, $.05/hr if bought from a small software firm, or $1 per hour if developed in-house and the cost is not amortized over a large number of users.

Thus the total cost of operating this CBI classroom averages to somewhere between $1.40 to $2.35 per hour,depending on the source of the lessons used with the system. This does not include faculty supervision.

The initial price is another point to consider. While many institutions might not balk at the large initial price for a 32-terminal network, they may object to the large maintenance costs associated with such a system. Others may not be able to purchase a minicomputer network, but could buy standalone CBI stations one or several at a time, and then later incorporate them into a cluster network at little extra cost.

APPENDIX D: Late Developments

CBI products now being tested or developed which may be announced by publication date are cluster controllers for two standalone CBI systems, another entry in the CBI microcomputer field, and a TUTOR-like language for hobby computers. This latter might conceivably radically alter an institution's choice of CBI hardware.

FOOTNOTES:

[1] "The Computer in Physics Instruction", Commission on College Physics, Nov.1965.

[2] Raster Monitor Evaluation and Selection, Conrac Engineering staff. Digital Design, 40, Jan. 1982.

[3] "Ticcit". M.D.Merrill, E.W.Schneider and K.A.Fletcher. Educational Technology Publications, Englewood Cliffs,NJ, 1980.

[4] "PLATO", H.F.Rahmlow, R.C.Fratini, J.R. Ghesquiere. Educational Technology Publications, Englewood Cliffs,NJ, 1980. Also see many publications by Computer Education Research Lab, 103 S.Mathews Av University of Illinois, Urbana IL 61801.

[5] An upgrading of the old IBM Coursewriter system. See PHOENIX, Appendix B.

[6] The RC-1 evaluated has been replaced by the RC-2 at lower cost. See REGENCY listing in Appendix B.

[7] After several model and name changes,the MicroPLATO station has emerged as the CD110 which can be used either as a PLATO terminal, a MicroPLATO station, or a CP/M operating system microcomputer.

[8] Local Communication Networks for CBI Systems. Donald L. Shirer. Proc. IEEE Conference on Computers and Communications, Phoenix, May 1982.

[9] The author system mentioned is sold by TERAK and licensed by the University of Utah.

[10] CAI Evolution: Mainframe to Micro, Paul Tenczar. Proc. Intl. Conf. on Cybernetics and Society, Oct. 1981.

[11] Selection of a Computer System for Computer-Based Instruction in Physics. D.L. Shirer. Proc. AAPT/AAS Winter Meeting, San Francisco, Jan. 1982.

[12] The order of listing after the first three systems mentioned is not meaningful. Several systems in lower groups can outperform CBI systems run on much larger computers. See the conclusion section for general recommendations.

# DESIGNING LANGUAGES FOR MICROS

George Gerhold - Western Washington University

Larry Kheriaty - Western Washington University

## ABSTRACT

Design of languages for micros is a multi-step process involving selection of worthwhile projects, knowledge of users and applications, and an iterative cycle of definition, structuring, reduction, extension, and testing. We will describe our approach drawing examples from our experiences with data-base management, ten versions of PILOT, and a new turtle graphics language.

Our title spawns a number of questions. First, why should ADCIS members be interested in language design? (That ADCIS members should be interested in micros is beyond question.) Many of you have been involved in some aspect of the creation of author languages or author systems; a comparison of your methods and our methods may be of interest. Most of you are involved in the selection of languages for educational applications, and many of the problems the language designer must solve are the same problems the language selector should be posing. A second question is: are there enough general principles to make this talk worthwhile, or does the design of a useful language involve such a high proportion of creativity and intuition that useful generalization is not possible? Our answer to this question is optomistic but not certain; we hope that you will find our generalizations and examples worth consideration. And finally a third question: what credentials do we have to discuss this topic? Our remarks will be based on our experiences with the design of a number of languages of interest to educators and trainers, a data-base management system called RMS, ten versions of COMMON PILOT, and a new LOGO-like graphics language tentatively named TURTLES.

A language design project starts when we realize that there is something worthwhile that can be done with a microcomputer, but that there is no language for doing that something. As an example, a few years ago a student at Harvard realized that a microcomputer could be used to remove the drudgery involved in preparation of business projections. At that time no suitable language for the application existed. The realization triggered the creation of one of the most successful pieces of microcomputer software, Visicalc. If we had a method for generating ideas of the originality of Visicalc, we would tell you what it was. However...

One can generate ideas by looking for extensions of existing applications. Possible extensions include improvements of existing languages, implementations of languages on a new class of hardware, revisions of existing languages to make them appeal to a new type of user, and, most frequently, some combination of all three. Our first version of PILOT was an improvement to the existing language CORE PILOT which also moved to a new class of hardware (minicomputers) and to a new type of users (heavy users of CAI at the college level). Later versions were implementations on new classes of hardware (micros) which also moved to a new type of users (elementary and secondary schools). One recent implementation of PILOT, the TRS-80C cassette version, is a simpler version for use in the home with minimal hardware. Data-base management systems of great power and sophistication have been available on larger computers for years. The RMS language was designed to bring the advantages of data-base management to a new type of user, a user without access to a programming staff and large computers, a user with need for some, but limited, hieratical structure in their files. Turtles, in ways that we will describe later, combines all three extensions.

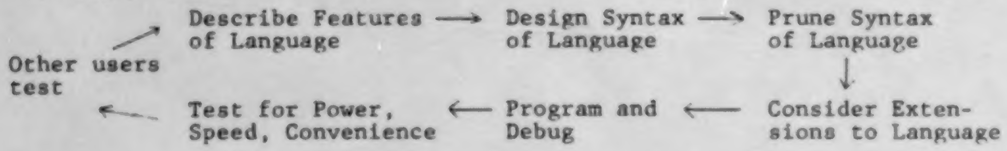
While designing and implementing new languages is fascinating work, it is work and not recreation. Therefore, the language designer looks forward to a payoff, at least wide usage of the resulting product and, preferably, money too. This means that the extensions must be significant enough to be obvious to the market. Our development process is an iterative one. Additional iterations might further improve the language but to little obvious effect, especially if in the meantime another -- and possibly inferior -- version of the language appears. Thus, in a language design project one answers questions of worth and significance at least twice, once in deciding whether a project is worth starting and one or more times in deciding when to quit refining.

At some point in the design process the hardware that the language is to run on must be selected. There are two things which must be considered here. One, the language must fit within the capabilities and capacity of the machine. For example, a cassette-based data-base management system makes no sense. Obvious, but there are text editors around with so many features that they can handle only 8K of text, and the currently available versions of LOGO for the Apple include so many features that the program space allows only about 2,000 keystrokes. Second, there must be a match between the marketing effort for the hardware, the interests of the potential purchaser of that hardware, and the potential applications. An APL enthusiast is not likely to purchase an Atari 400; a reduced version of APL for the Atari is not going to be a big seller! Our 6809 version of COMMON PILOT was designed for the SwTPC /09, S/90 series, but the manufacturer never got interested in the educational market.

For us the actual design and implementation merge in a cyclic process. We make a number of iterations through the cycle. The following diagram shows one of several possible logical arrangements of the steps.

We'll first amplify the descriptions of these steps, and then we'll illustrate our procedure by describing two projects. We've begun the whole process by identifying a potential application, in the typical case a significant variation from some existing language. Therefore, we can usually write an adequate description of the language by listing the changes we plan to make in the parent version. The formal description of the language is established by describing the syntax the user will employ and by defining the data structures which will be used. Compatibility with the parent language may be desirable, in which case much of the syntax is already established. Definition of the data structures is one of the critical (and once done, irreversible) steps in the process. The logical structures possible within the language will be determined by the choices (e.g., global variables don't mix with recursion), and the speed of the language will be a consequence of these structures. Because we are usually trying to make a micro do more than previously thought possible, it is difficult to overemphasize the need for system programming skill and experience at this point.

We believe that the normal evolution of computer languages follows a progression from a well-designed core through worthwhile extensions through less useful extensions to eventual confusion. Therefore, everytime we rework a language we ask what can be left out or absorbed elsewhere. Especially in the micro field, extension to new applications does not automatically mean adding features to a language; it often means removal of unnecessary "features" and complexity. This removal is what we mean by pruning the syntax. In this way we try to slow the slide towards confusion. At the same time one must look for truly useful extensions which will increase the utility of the language. Sometimes these are obvious; the integration of graphics and sound capabilities available on some of the microcomputers is worthwhile for instructional languages. In addition, we look for one or two simple additions to the syntax which have major implications for use of the language. This is difficult to do and to describe in general. We will return to this point in our examples. Of course we remember our


Describe Features of Language
Design Syntax of Language
Prune Syntax of Language
Consider Extensions to Language
Program and Debug
Test for Power, Speed, Convenience
Other users test

desire to prune the syntax while considering extensions. Programming and debugging require no additional description. Testing occurs in two stages once debugging is complete. We check for execution speed; we try complex programs (often discovering new features and flaws in the language); and we try to view the language from the point of view of the user. The second stage of testing involves other users. We've found that a few inventive and demonic testers are more useful than a large crew of anonymous beta testers who are mainly interested in free, pre-release copies of software.

Once we have completed one cycle, we begin again. This time the parent is our version 1, and the changes are based on the reactions to the tests. Of course the diagrammed sequence is not followed strictly. For example, we usually have many unstructured lunch table discussions of the first four steps before much gets written down on paper. After numerous iterations, formal and informal, complete and incomplete, we have two more difficult steps to complete. We have to write a manual, and we have to decide when to quit refining the language. Writing the manual is another form of testing; we find places where the syntax could be improved and where the language can be made more user friendly in the process of "explaining" the language to new users. Quitting seems to be more difficult for some people than for others. Perhaps our boredom threshold is low enough to avoid the problem.

Let's now turn to specific projects. We recently completed several versions of COMMON PILOT for the TRS-80C. The disk version is the more powerful PILOT, but the cassette version provides better examples of our design approach. This project was an extension to a new class of computer, a 16K home computer and to a new type of user, the home user. The parent language was COMMON PILOT. The major changes needed were eliminations, a full version of COMMON PILOT would not fit in the machine. From our experience with CAI we knew that high resolution graphics, color, upper and lower case text, string operations, and user defined characters were essential. The memory implications of this were such that everything else had to be pared to the bone. We had to live with single character variables and integer arithmetic. By rewriting the expression processor in COMMON PILOT (which effectively converts expressions into reverse polish) using a technique called recursive descent, we could save space. Multiple matches could be used in place of AND operators in the match instruction. These are a few examples of the decisions which were made informally by discussion of the first steps of the cycle before a description of the language was put on paper. Once an operative version was completed we had an accurate fix on the memory use, and we were able to restore a few very useful features. The memory space is so limited that the interpreter can grow only at the expense of maximum length of applications programs. We initially eliminated all features which could be covered by multiple instructions of the reduced version and then used our experience with typical applications programs to decide which restorations would allow one to do more. Basically, the question is will the additional space needed in the interpreter for the feature in question be greater or less than the space saved in the typical applications program by use of the feature.

Let's concentrate on the least routine parts of the process, the search for simple extensions with significant implications and the hidden benefits of structured programming. COMMON PILOT includes one of those simple extensions, the execute indirect instruction. This one instruction, which at the assembler level requires only a few additional lines of code, executes any string variable as a PILOT instruction. This leads to simple implementation of directories, custom designed calculators, randomly generated verbal problems, and a host of other features in PILOT programs. A somewhat larger addition, a simple screen-oriented editor, made the system much more user friendly. One addition to the editor, suppression of the carriage return for lines ending in @, allowed use of the language as an extremely simple (and cheap) text editor. The combination of the execute indirect and the text editor allowed us to add an immediate mode -- in which the user can experiment with individual instructions -- by the inclusion of a three-line, twelve-character program called forth by two keystrokes. The use of recursive descent for the expression processor made it easy to implement auto-conversion between strings and numbers, thus simplifying what has been a troublesome point for users. In no case were all of the consequences of these extensions obvious when they were proposed; they only became so during the repeated iterations.

Our current project is the development of a structured turtle graphics language tentatively named TURTLES. The parent language in this project is LOGO.

We are both adding to and pruning the parent language; we are moving the language to lower cost hardware (the TRS-80C) and appealing to a new type of user. We thought that the existing micro versions of LOGO were out of balance with the hardware and with the users. So many features were included that there was insufficient space left for user programs; much memory was used to provide list processing of strings, but most users were only using the turtle graphics. By elimination of string manipulation and by redesign of the data structures we could save enough space to fit it on a 32K machine with room for extra features. Again in this case we used informal iterations through the early steps of the process before putting pen to paper.

LOGO is a language whose major use probably will be in the elementary grades. Therefore, a simpler implementation on lower cost hardware is likely to be successful. But again we wanted to look for any simple additions which would extend the capabilities of the language. We felt that any extensions should be consistent with the philosophy behind LOGO. We have made two such additions and are considering a third. We had some indications that an artificial lower age limit for use of LOGO was established by the use of key-words for the primitive actions of the turtle. We hope to reach a pre-reading audience by including a symbolic (rebus) mode of programming. Procedures can be defined by pressing symbolic keys instead of by typing keywords and numbers. LOGO is an attempt to introduce important advanced ideas at a very simple level. Additional ideas like multiprogramming and message passing via mailboxes can be introduced via multiple turtles. Multiple turtles is a simple addition with many consequences; in addition to multiprogramming etc., games, user-created animation, and strategy testing all are possible. (We must admit that multiple turtles is a simple addition for the user, but not for the systems programmer.) We have concluded that the consequences of the multiple turtles are so great that we could call our version something other than LOGO, something which emphasizes the multiplicity, hence TURTLES.

Let us conclude by summarizing. In preparing this paper we discovered that there was indeed some repeated system underlying what at the time seemed like a completely chaotic process. We think that there are five components to whatever success we have had as language designers. One, the small number of people involved in the process led to a tight iteration loop which was repeated many, many times. Second, the language designers are also users of the language, a process which aids especially in the pruning phases. Third, the systems programmer is highly skilled and experienced. Fourth, we have been able to come up with simple but very powerful additions. Fifth, in most cases we have been able to write the manuals, so that the manuals are written by people who are both the designers and heavy users of the language. And finally, because simplicity itself is a worthwhile goal, we continuously compare what we have with the goals of the language in order to avoid inclusion of unnecessary complexity. Remember that the reason for creating a special purpose language is to solve a particular limited class of problems well.

# COMPUTER-ASSISTED EDUCATION:

# WHAT'S NOT HAPPENING?

Francis D. Fisher
Henry Luce Professor of
Ethics and the Professions
Haverford College, Haverford, PA 19041
(215) 896-1070

## ABSTRACT

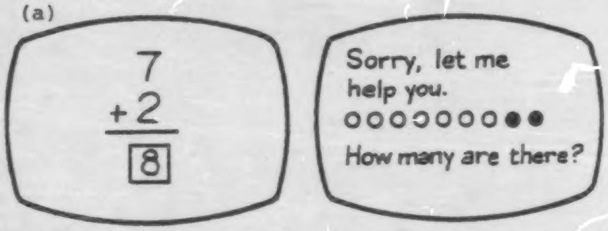
Why is there not more good computer courseware about? In the summer of 1981 the Ford Foundation supported an outside investigator in asking that question of teachers, professors of computer science and cognitive psychology, computer manufacturers, "courseware authors," and educational publishers. Their answers indicate that:

> The educational courseware beginning to appear is mostly simplistic in design and fails to exploit the full teaching power of the computer.
>
> There is inadequate expression of the pedagogical principles unique to the use of the computer and what is known is inadequately communicated to or utilized by courseware publishers.
>
> School use of computers is not restrained so much by teacher fears of being replaced or problems of cost as by poor quality courseware and the difficulty of schools' adjusting to a different teaching system.
>
> There is a danger that computer-assisted learning will bypass the public schools and take place only in the homes of the affluent and elsewhere on a commercial basis.
>
> The solutions to these problems will probably not come about soon by operation of the free market alone.

While few controlled evaluations have as yet been completed on courses of computer-assisted instruction, examples of excellent computer courseware are persuasive that this interactive device can be an individual tutor of infinite patience, without bias, inviting to use and powerfully effective. BUGGY developed by John Seeley Brown can diagnose a multitude of newly identified patterns of student error in simple subtraction and offer instruction targeted to aid the individual. DARTS, developed by Sharon Dugdale of the University of Illinois, engages the student in shooting at targets in a way which directly increases an appreciation of the relative value of numbers other than integers; the motivation to learn is inherent in the excitement of the very task that educates. In a program integrating the computer with laser-read videodisc, WICAT has developed for the Smith Kline corporation a series of simulations of gastro-intestinal patients, by which physician-students can question, test, and practice alternate diagnoses and have their skill critiqued. Seymour Papert of MIT convincingly demonstrates that young children can find in the computer an environment for learning by programming which gains for them difficult methematical skills in less abstract and more personal-related explorations than by conventional curriculum.

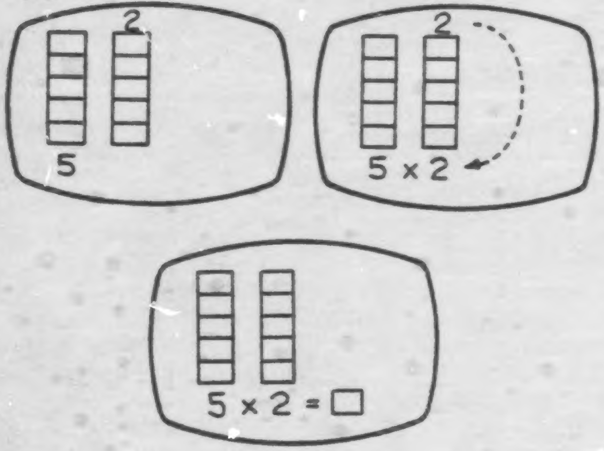
There are, of course, some other examples of excellent educational use of the computer, but what is remarkable to the new viewer of computer courseware is how much of it is poor. Most of the courseware now available does not use the creative techniques and exploit the medium as do the examples above. Sound and motion are too often used in ways which may attract a buyer but distract rather than concentrate the attention of the learner. Also, bits and pieces of courseware are being sold which do not fit together or into existing curricula. Worse, commercially available courseware often reflects poor pedagogy. Some of it is no more than electronic page-turning; little of it incorporates the insights into the special pedagogical possibilities of the new technology which the best examples demonstrate. Consider the following four illustrations of basic math materials drawn from the 1981 product lines of three of the largest commercial publishers of computer courseware.

(a)



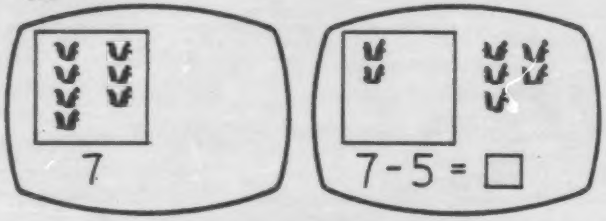
In example (a) the child has given a wrong answer to the problem presented. The program responds with the display on the right, which is, ostensibly, an attempt to instruct the child. In this case, the directions to the child are inadequate. The child is asked, "How many are there?" But how many what? Empty circles? Dark circles? How many all together? The child cannot reasonably be expected to know what is being asked.

(b)



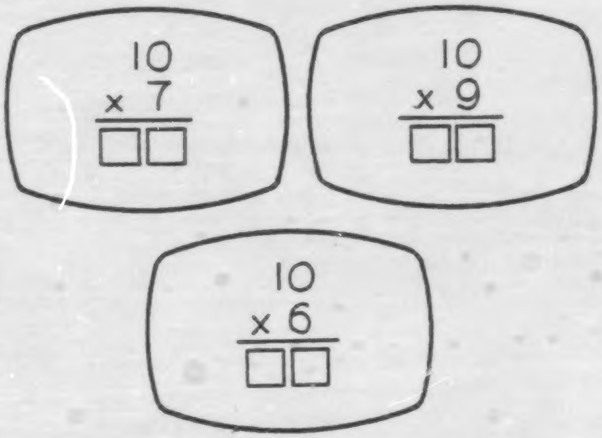
In the second frame of example (b), the "2" migrates (to the sound of electronic music). Careful observation of this motion will not help the child in any way. This is a misuse of action and miscues the child in that attention is drawn to the wrong thing. Indeed a powerful lesson is taught: Careful observation is worthless.

(c)



Example (c) shows misuse of spatial arrangement within the display. Five of the seven squirrels in the box at the left scamper out (as seen on the right). The two squirrels (which presumably the child is supposed to count) are over the seven. The box for the answer (2) is placed under the five squirrels.

(d)



Example (d) illustrates yet another problem, thwarting a natural progression in the child's learning. To type in the two-digit product, the student must first type the digit for the units column, then the digit for the tens (e.g., in the left frame 0-7). By the third example, many children will see the pattern and want to type in "60" (6-0), but they will be wrong for they must type the "0" first.

## REASONS FOR COURSEWARE FAILURES

These failures in simple programs make clear that commercial publishers, those who are providing the standards of today's marketplace, are far from appreciating the powerful teaching possibilities demonstrated by the few individuals who are experts on computer-assisted learning.

The problem seems to be that most of those with the best ideas of how the new technologies might be used work in research laboratories at universities, such as at the University of Illinois, University of California at Irvine, and Massachusetts Institute of Technology, or within the research divisions of electronics firms, such as the XEROX laboratory at Palo Alto. Researchers are often academic in outlook and following the incentives of higher education are not so concerned with courseware production, whereas those involved in commercial publishing seem unknowing or unconcerned with the subtleties of the new pedagogical possibilities or excessively concerned with quickly producing a marketable product. The researchers are not getting out information on their state-of-the-art and commercial producers are not producing courseware which reflects that knowledge and art.

It should perhaps be made clear at this point that this discussion does not apply to the small, straightforward programs which an individual teacher might prepare for use with his or her own class. Such small programs are analagous to the teacher-authored handout of a mimeographed paragraph or an example drawn on the chalkboard. As teachers increasingly gain some grasp of programming, as they certainly will within the next generation, they will be able to exploit the computer for such limited purposes.

The production of a large programmed course of study, on the other hand, demands not only subject matter expertise but also the skills of sophisticated programmers; not only a general pedagogical approach of the sort any teacher might be hoped to have, but also knowledge of the special pedagogical possibilities permitted by the new interactive, computer-based technologies. Substantial investment of time must be made years before a large sales network begins to return money. And after production, the course must be "maintained" by correcting newly discovered errors and adjusting for improvements in the evolving technology. Indeed, in scale and complexity the effort required is more closely analogous to producing a movie than to writing a textbook. Such a costly undertaking, requiring varied talents, is a job for a large organization. Two sorts of large organizations are currently producing educational courseware: publishing houses which specialize in educational materials and computer manufacturers.

The educational publishers seem to assume that their experience in supplying printed textbooks to schools equips them to produce courseware for the new electronic technology. With print, however, the publishers generally commission an author or receive a manuscript from someone who is familiar with the pedagogical principles involved in both the presentation of the subject matter and the use of print media. With electronic courseware, publishers do sometimes receive submissions from the new cottage industry of courseware authors. But in the main these are small efforts; programming a substantial work is generally beyond the resources of single individuals. So the publishing firms have, by and large, become authors. However, as authors, they tend to focus on getting their products quickly on the market to meet the demands of a computer-hungry public before they lose their place in the education market. As a result, they frequently end up by just putting their own textbooks on line. The result is similar to the early efforts at educational TV when the programs consisted of a teacher sitting in front of the camera, with no exploitation of the medium. The result was a talking head. Likewise, if you program a textbook onto a computer, you end up with a moving book.

Computer manufacturers, on the other hand, have a good grasp on how the hardware can be used for storing data and performing large computational tasks, the purposes for which the machines were originally designed, and they are interested in exploiting the medium to the extent that they can show off what they understand to be the capabilities of their hardware products. But the hardware manufacturers seem even less likely sources of sound pedagogy. In fact, computer manufacturers are likely to follow the market rather than to show the way themselves. (One computer executive described courseware development as being "in response to the market," explaining, "After we get the market report, we write up the product specifications and turn them over to the production people," as though schools knew what might be done with the machines.)

In some instances, computer manuturers have combined with publishers, but it appears that in these cases the combination is greater than the sum of the parts only in the area of sales. While educational publishers are often assumed to be a natural source of computer courseware because of the close relation of their textbook sales force to the schools, publishers dare not distribute courseware alone without the help of computer technicians since the customer uses the electronic courseware together with a machine and wants but one telephone number to call when the combination isn't working. If educational publishers bring little understanding of the pedagogical possibilities of the computer, are unfamiliar with its technology, and are not using their own sales force for selling courseware, why are they the expected major source of computer courseware? The answer at the moment seems to be based more on expectation than good reason and an absence of strong competition from elsewhere.

Some aver that the large front-end costs of producing good courseware have created a "chicken or egg" situation in which the publishers fail to invest in making good courseware because the demand is not there and the schools fail to invest in the necessary hardware because the supply of courseware is poor. The large expenditures which are already being incurred in producing courseware tend to belie this, although it is true that no one is developing courseware for the technologies still on the drawing board. Also, exaggerated as reasons for the scarcity of good courseware, I believe, are the fact that programs developed for one type of computer can't be run on others and the difficulty of assuring protection of intellectual property through copyright or otherwise. The problem of compatibility seems to be shaking out, and courseware can now be protected from all but the zealous individual code-breaker.

The most compelling explanation for the scarcity of good courseware seems to lie in the shortage of people with expert knowledge of the pedagogical possibilities of the new media and the weakness in communicating to producers such knowledge as there is. The publishers, we hope, are not intentionally producing second-rate material. Perhaps they do not know what the best is.

Unfortunately, the standards may be set within the next few years. We should be frightened by the example of printed textbooks, in which publishers compete within a fairly narrow band of almost uniform mediocrity. What we urgently need, and soon, are examples of courseware which compellingly demonstrate to producers, schools, and society the exciting contribution that the new technology can offer to education.

## SUPPORT FOR RESEARCH CENTERS AND INFORMATION EXCHANGE

The well established and productive centers of knowledge about computer-assisted instruction must continue to be supported, no easy task as National Science Foundation interest is reduced. These centers not only produce much of the needed knowledge base, they also constitute the critical mass of scientific activity necessary to attract and train additional scientists now so tempted by commercial job opportunities.

Information about developments in computer hardware is now rapidly diffused. But an exchange of information about the educational use of the computer needs stimulation and support. We could gain from better information of four kinds:

1. Pedagogical ideas now known. There is a need to support the accumulation and publication of what principles are now known about how to design good courseware. This material is sprinkled about in journal articles of various disciplines, conference proceedings, and unpublished knowledge in the research centers.

2. Software used in producing courseware. Used in the production of successful computer-assisted instructional materials are specific software programs developed by authors to make programming easier (e.g., how to represent a molecule on the Apple computer). This emerging collection of "machine tools" should be investigated to see how the inventions might best be shared.

3. Collections of courseware. The market of courseware is impeded by the difficulty of making a decision to purchase a computer program on the basis of a description appearing in print. Evaluation processes and journals reviewing courseware are being established. In addition a few centers need to be located across the country where large collections of courseware can be seen and experienced by other authors and potential users.

4. The development of a national agenda for research in the relationship of learning and the new technologies is itself an important activity that could be improved, an exchange of views on what research is in prospect, what might be undertaken and who is interested in doing what.

## PROBLEMS FOR SCHOOLS

If good courseware is produced, will schools buy and use it? Some publishers believe schools will fail to embrace the computer because teachers are afraid they will be replaced by it. Anticipating this fear, cautious publishers promote materials for courses for which teachers are in short supply, like engineering, or for subjects which are new to the curriculum, like computer literacy. They market their courseware as "supplementary" to existing classroom routines and as a means of relieving the teacher of the tedium of drill or recordkeeping so as not to strengthen teachers' fear of being replaced.

But these publishers' concerns may be excessive. Teachers seem more worried about the need to learn a new skill than about losing their jobs. Indeed, many teachers are excited by the new technology; most believe it is coming whether they want it or not; and a few have already begun to gain status in their own schools by becoming computer "buffs" to whom other teachers turn for help. Further evidence of teacher acceptance comes from the fact that teachers are over-subscribing those courses which are being provided to introduce them to the computer and that they are listening attentively to their computer-wise students.

Procurement decisions, moreover, are not made by teachers but by administrators, who are not worrying about their jobs but about their budgets. So just how expensive are computers?

Arthur Melmed of the National Institute of Education estimates that 30 minutes of individual instruction could be supplied daily to each elementary school child by 1990 when the cost per child for both hardware and courseware would be about $25 per student (Melmed, 1982).

All in all, the cost of computers, like the fear of teachers losing jobs, should not be seen as a barrier to computer-assisted education. It is more likely that the rapid spread of computers in schools is now inhibited by two other factors, the absence of good courseware referred to above and the problems a school would face if computer-assisted education does increase the efficiency of the instructional process.

Efficiency in teaching is welcomed in industrial training programs, computer manufacturers point out. "If a sales training program that used to take eight weeks requires only two weeks with computer-assisted methods, any sales manager can figure to the penny how much he has saved. What is the matter," they ask, "with the schools?"

Well, if an algebra course that used to take 16 weeks takes only five with the computer, the school has not a saving but a problem: What to do during the other eleven weeks allotted to the course? It seems that the tasks we have assigned to our schools are measured not in educational output, but in the day-care terms of hours to be spent inside a school building. Of course the time saved in learning algebra could be put to use learning some more of the many things there are to learn, but expanding a curriculum has costs.

If the educational promise of the new technology is to be realized, society must reassess the role of schools and either relieve them of students who have learned "it," shifting the problem of their occupation elsewhere, or support an expansion of the curriculum. Over many years, savings in instructional manpower from computer use may meet the expenses of teaching additional subjects. But in the short term, an increased investment in education will be necessary if schools are both to occupy students and to take advantage of the possibility of increased productivity in education offered by computer use. We can teach more and better, but we will have to pay more. The issue of how schools might best adjust to the arrival of efficient computer-aided instruction is not receiving the attention it deserves.

The prospect depends on good courseware. The more likely scenario, unfortunately, is that the courseware available to schools will continue to be weak. The schools will either recognize the deficiency, feel betrayed, and consign computers to join a collection of oversold audio-visual machinery in some dusty closet; or, perhaps more likely, the schools will adopt the existing courseware programs and fail to recognize that computer courseware falls far short of its potential.

## OUTSIDE OF SCHOOLS

Computers are entering homes more rapidly than they are being put into schools. The individual purchaser has business, professional, and entertainment purposes primarily in mind; educational use is an extra. But the home educational market for software is promising. One toy manufacturer, for example, has sold one-and-a-half million sets of a computer-driven toy designed to help young children master spelling.

The vast majority of these toys were sold not to schools but to individuals, confirming that parents are prepared to pay for educating their children directly, not just through the payment of taxes to support the local schools. The home market, moreover, is reachable by mass marketing techniques that move sales faster than the formal procurement processes of school districts. Designers of courseware for home computers may be less constrained than sellers to schools in combining entertainment with instruction, for better or worse.

While we cannot yet predict how soon good courseware will reach the home, we can be certain that children in homes with computers are apt to gain skills of computer literacy and the intellectual growth that comes from learning to program. It is not unlikely that computer-assisted education will thrive in the home environment, leaving behind the public schools and less affluent children, thus increasing the gap between them and the well-to-do that already threatens our social stability.

It is possible that institutions other than the schools will fill this gap and bring the benefits of good computer-assisted education to an audience outside the homes of the rich. The Children's Television Workshop, producer of Sesame Street, is exploring the educational use of computers in arcades, shopping centers, and learning parks. The British Open University has set an example of how public television can teach all, regardless of wealth, and is now equipping regional centers with computers and courseware.

## CONCLUSION

The great educational potential of computers is not likely to be realized in the short term through the operation of free enterprise alone. The education market may be huge, but it will not demand, and commercial publishers are not likely to produce, a product of better quality than that which is widely known and appreciated. Hence those charged with public interests must help realize the promise of good courseware.

Even were the character and value of good courseware well understood, there would be a need to involve our public institutions in research, development, and dissemination. For computer-assisted education is no different from education in general (and unlike industrial training) in that its value accrues in large part to the benefit of society and can only rarely be captured in full by the provider and taken as compensation for a better teaching effort. The social returns from computer-assisted education call for its promotion by social institutions. But which ones?

Individual local school districts lack the competence and resources to undertake the necessary research and development effort. Some states are large enough and a number are involved through state universities or agencies such as the Minnesota Education Computer Consortium. But the unevenness of states in their resources and the clear nationwide benefits at stake call for a substantial federal effort in the inherently public function of supporting research and development in education technology and in distributing information about what is known. Without such an effort, the optimum allocation of state and local resources alone will not result in the pattern of effort that is optimum for the nation.

An analogy exists in the field of health technology, where the National Institutes of Health supplement drug companies, hospitals, and doctors. The need for a federal role in education technology is recognized by Secretary of Education Terence Bell, who is sensibly seeking to have the Department of Education assume the appropriate federal role in educational technology, a role which the National Science Foundation is seeking to cast off.

That is not to conclude that we can dump the task of improving computer-assisted education into the federal lap. Indeed there are roles for all--local schools, states, universities, publishers, computer manufacturers, and public-concerned agencies like foundations and public television stations. Gaining a better understanding of who is doing what and deciding who should be doing what are first priorities for discharging our responsibilities to the computer-assisted generations to come.

## NOTES

Arthur S. Melmed, "Information Technology for U.S. Schools," Phi Delta KAPPAN, 63 (Jan. 1982), p. 310.

# SOFTWARE -- HOW GOOD IS IT?

Florence M. Taber, Ed.D., Interpretive Education

Alonzo E. Hannaford, Ed,D., WMU

## Abstract

Microcomputer software is now available for instructional uses. Unfortunately, the importance of instructional software is frequently overlooked by those who are considering the purchase of microcomputer hardware. Too often the microcomputer is purchased and then is not used for effective instructional purposes because too much of the available software is not based on sound educational principles nor does it use the capabilities of the microcomptuer to best educational advantage.

The key to effective software is evaluation because if close scrutiny of educational software occurs before purchase, software producers will be forced to develop what the market demands.

The first step in software evaluation is to determine the cognitive, affective, sensory-motor, curriculum and learning needs of those who will be using the software. After these needs are developed, they should be used while systematically scrutinizing software using a specific form that is based on these needs. The various aspects of educational software which should be evaluated are Instructional Content, Educational Adequacy, and Technical Adequacy. Without considering all three of these qualities, educational software will not meet the needs of learners -- or of teachers.

---

Microcomputer software is becoming more and more available to be used for instructional purposes in the classroom, in industry, and in the home. Unfortunately, the importance of instructional software is frequently overlooked by those who are considering purchase of microcomputer hardware. The initial glamor lies with the hardware itself; and it is not until the actual need arises to put the hardware to use that it is realized that, without software, the best microcomputer hardware is worthless. The general emphasis which has been placed on development and selection of software in instructionally-based computing belies its importance. Far too much of the available software is not based on sound educational principles nor does it use the capabilities of the microcomputer to best educational advantage. The technology behind the development of hardware has, unfortunately, far outpaced that surrounding the development of educational software.

The literature supports the indication that there is little educationally sound software on the market. Isaacson (1981) reported that little effective software is available; some exists but revision is frequently necessary before it is technically and educationally adequate for instructional purposes; and many of the programs are poorly designed, have not clear-cut objectives, and contain "bugs." Much of the literature also supports the opinion that prior to hardware selection, software evaluation needs to be implemented because software is hardware dependent and the programs that meet instructional needs must be compatible with the available microcomputer (Bell, 1981; Fenzel, 1981; Hannaford & Sloane, 1981; Taber, 1981; Thornburg, 1981).

The potential effects of this state of affairs are as yet unknown but one apparently viable scenario is that because of the durth of educationally adequate software, many users will become disenchanted and the current impetus behind the "computer in education" movement will drastically wane just as did the enthusiasm found in the 1950's and 1960's. Unlike previously, however, the interest will not wane because of a lack of hardware or the cost of the hardware but from lack of sufficient high quality software. While this is but one scenario, past practices with other educational media tend to lend it some credence.

Sound educational software can and will be developed if there is pressure to do so. Close scrutiny of educational software before purchase is necessary. If informed evaluation by consumers occurs, producers, who fail to develop sound educational software, will be forced out of the marketplace. Producers of software, just as in all other areas, will develop what the market demands. It is thus essential that potential purchasers ask the question "Software -- How Good Is It?"

In order to evaluate software, the first step is to determine the needs of the user. Any educational material that does not meet the needs of the user is useless to that individual. Cognitive needs, affective needs, sensory-motor needs, curriculum needs, and learning needs are necessary considerations. Such determinations are an essential first step since they circumscribe, to a large extent, the software needed.

After the needs of the learner have been carefully determined, a search for software designed to meet those needs can begin. Evaluation of software must be systematic due to the complexity and depth of information necessary to facilitate the decision-making process. One way of systematizing this process is through the use of a form which contains items to be considered. Such forms can be designed by the user for the specific situation or can be selected from those found in the literature (Holznagel, 1981; Hannaford & Sloane, 1981; Taber, 1981). The various aspects of educational software which should be evaluated are Instructional Content, Educational Adequacy, and Technical Adequacy.

## Instructional Content

After it has been determined that the software does address the desired content area or areas, a number of other considerations need to be addressed concerning that content. Since, unlike print media, the consumer cannot scrutinize the specific content without actually interacting with the program, the goals and objectives of the software need to be specified as do each of the concepts to be presented. There likewise should be a method by which each objective will be evaluated as to the learner's success in the program. An instructional guide should also be available which:
1) Identifies the prerequisite skills, vocabulary, concepts, and experiences;
2) Describes how to present the program;
3) Outlines the content presented; 4) Contains supplementary materials as necessary.

## Educational Adequacy

A primary consideration in evaluating software is determining whether or not the computer is the most appropriate medium given the learner, the content, and the specific learning environment. Some software is nothing more than an "electronic book" with the only interaction being the "turning" of electronic pages by the user. Clearly the appropriateness of such software is highly questionable. The cost-effectiveness of this medium must also be a consideration when determining the appropriateness. All -- in -- all, the reasons FOR using this medium must outweigh the reasons AGAINST -- if it is to be appropriate.

The area of educational adequacy also includes how well the software makes use of accepted teaching principles. For example, the content should have been task analyzed into appropriate and sequential steps. Responses to question or interaction frames should be clear and easy to understand. Frames following question frames should be reinforced either by providing feedback as to why the response was correct or by providing additional information if the response was incorrect. Unfortunately, too many programs on the market provide punishment following incorrect response (i.e. "Dummy!"). The sentence length and vocabulary used should reflect not only the reading and conceptual levels of the probable user, but also the user's interest level. Branching should also be used to individually move the learner to higher or lower instructional levels based on learner responses or on teacher discretion. When a learner makes an error, the branching should provide for a re-teaching of the concept in a different manner and should present the concept in smaller instructional steps that was previously done.

Evaluation should involve determination of the appropriateness of the type of instructional software for the users. For example, most programs are drill and practice: they are relatively easy to program. Others are tutorial, simulations and/or games, discovery, or problem solving in nature. Each of these types has applications where its use is appropriate and applications where it is not. The same can be said for its use with various groups of learners so this must be taken into consideration.

### Technical Adequacy

Programming considerations also need to be evaluated. The formulating of each frame is especially important. For example, in most situations, lines of print should be double spaced with six to eight lines presented on the screen at a time -- depending on the type of microcomputer. The important thing to remember is that the text or graphic should be centered and should not look cluttered. Graphics, color, animation, music, inverse right (black on white), blinking/flashing information, etc. should be employed as an attention getting and directing device when appropriate. They can enhance the material educationally but care must be taken that they are not overused for they can also be very distracting.

The program should run to completion without errors that stop it and/or cause the computer to interrupt the program and go into a different mode of operation. Good educational software will also be able to prevent inadvertent interruption by the user such as can occur by hitting "break" or "reset" keys. It is possible to prevent such actions from exiting the user from the program. It is also possible to handle inappropriate input in such a manner that the program is not interrupted. Good software should do all of this.

A primary educational factor which falls within the purview of technical adequacy is the source and amount of control afforded by the software. Control may be totally internal, meaning that such things as content, rate of presentation, use of sound, rate of reinforcement, etc. are completely controlled by the software. This is what is generally found in educational software. The learner goes where the program leads or does not go at all. On the other hand, software can be designed to afford the teacher and/or the learner varying degrees of control. For example, the teacher can be allowed to "tailor" the program for individual learners by specifying such things as entry level, sequence, speed of presentation, style of type, type of feedback, rate and type of reinforcement, and use of features such as color, sound, etc. Such flexibility means that the teacher can actually "direct" the program to do what is desired -- a powerful teaching tool indeed. Students, too, can be afforded a certain amount of control of the software. This might include control of pacing of the speed at which the material is presented, control of point of entry into the educational module desired, allowing review of previously covered materials, skipping unwanted or already completed material, getting additional explanation or help when needed, exiting the program when deisred, leaving "messages" or questions for the teacher, etc. While it is not necessary or even appropriate for all educational software to have these features, for certain types it is necessary if they are to be of high technical quality and maximum educational benefit.

Therefore, if educational software is to be of good quality, it must demonstrate good Instructional Content, Educational Adequacy and high Technical Adequacy. Without such qualities, educational software will not meet the needs of learners and teachers; and without meeting these needs, the technology surrounding computers in learning will not be used.

### References:

Bell, A.H. "Computer anxiety." Media and Methods. 1981, 1(5),28-29.

Frenzel, L.E. "Learning with micros." Interface Age. 1981, 6(12),44-5.

Hannaford, A.E. & Sloane, E. "Microcomputers: Powerful learning tools with proper programming." Teaching Exceptional Children. 1981, 14(2), 54-57.

Holznagel, D. "Which courseware is right for you?" Microcomputing. 1981, 5(10), 38-40.

Interpretive Education, Inc. MCE Evaluation Form. 157 S. Kalamazoo Mall, Kalamazoo, MI., 1980.

Isaacson, D. "What's holding back computer use in education?" Classroom Computer News. 1981, 1(5), 28-29.

Thornburg, D. D. "How to select a personal computer?" Recreational Computing. 1981, 52, 10-12.

# Computer Literacy in Business and Industry

## Experiences With Microcomputer Workshops

Greg Kearsley

HumRRO

There is a tremendous need for computer literacy in the business world. While most corporations have spent a large amount of time and money developing training programs around specific computer systems and applications, little has been done to provide employees with a general understanding of computer technology and how it can be used to improve job productivity. The only exceptions are "high technology companies" (e.g., Clogston, 1980) where the investment has relatively obvious payback. Recently, a number of managers, professionals and entrepreneuers have explored microcomputers to give themselves a productivity edge over their colleagues and competitors (Seaman, 1981).

As part of HumRRO's broad program in computer literacy (e.g., Seidel, Anderson & Hunter, 1982; Hunter, 1981), we have developed computer literacy workshops for the business/industrial domain based on microcomputers and off-the-shelf software. These workshops represent an attempt to explore the computer literacy needs of the business world. We have focused on what we feel are three critical components of the business community: managers in middle/large corporations, principals/employees of small business, and training/designers/developers. This article describes the three workshops and our initial experiences with them.

## Micros for Managers

Microcomputers are quickly becoming an important management tool. Thousands of managers in all types of organizations have acquired and are using microcomputers to assist them in their decision-making, problem-solving and planning activities. Typical application areas include project management, profit models, data analysis, budgeting, financial forecasting and information retrieval. Seaman (1981) describes a number of actual examples as well as discussing hardware and software considerations for management applications.

The major goal of the workshop we developed was to demonstrate to managers how microcomputers could be used in their job activities. The seminar was built around three typical management tasks: project management, budgeting, and using personal files. It attempted to show managers how to perform these tasks more effectively through the use of a microcomputer. By means of hands-on experiences, we wanted the managers to make their own judgments about how valuable microcomputers would be in their work.

For this and the other workshops, we used a popular microcomputer (the Apple II Plus) and "off-the-shelf" software (Visicalc, Apple Plot, and CCA Data Base Management System) which can be bought in almost any computer store. We made it clear to the managers that they could buy everything they were using "as is" and did not need the assistance of their data processing departments. We also made it clear that we were not endorsing the particular hardware and software being used in the workshop and that other brands would have done just as well for our purpose. (This is true in this report, as well.)

The workshop outline is shown in Figure 1. In the first module, we introduced the managers to the components of the system and demonstrated the essentials needed to use it (e.g., how to power on and off, inserting diskettes into drives, loading program and data diskettes, etc.) We did not teach anything about the internal workings of the hardware/software (no bits, bytes, or bauds) nor did we even talk about computer languages. It is our feeling that initially a user should be in-

Figure 1

MICROCOMPUTERS FOR MANAGERS WORKSHOP

**Module 1: Introduction**
- Components of a microcomputer system
- How to use the system

**Module 2: Project Management Example**
- Using Visicalc to develop a project staffing chart
- Using Apple Plot to display project staffing data
- Using CCA DBMS to store and retrieve staffing data

**Module 3: Budgeting Example**
- Using Visicalc, Apple Plot and CCA DBMS to develop, display and store an annual budget

**Module 4: Information Retrieval Example**
- Using CCA DBMS to create an electronic filing system

**Module 5: Wrap-Up**
- Other applications
- Strengths/Limitations of Micros
- How to select and buy a system
- Further information (computer stores, magazines, books)

troduced only to the small amount of information necessary to actually use the system. After all, we do not try to turn automobile drivers into mechanics nor stereo owners into electronics technicians and the same should hold for computer users.

The second module involves a project management example. In this example, we briefly introduce the application (tracking project staffing costs) and then show how Visicalc (an electronic worksheet) can be used to set up and make changes to a staffing chart over the course of a project. Following the Visicalc demonstrations, we then show how the Apple Plot program can be used to display this data in the form of bar charts and graphs. Finally, we demonstrate how the CCA program (a data base management system) can be used to store historical project staffing data which can later be retrieved for comparison with current staffing figures with either Visicalc or Apple Plot. After the demonstrations are concluded, the managers are then given a problem sheet (incorporating step-by-step job aids) which requires them to use the three programs to make simple changes to the examples they have already seen (e.g., adding or deleting some project members, plotting a subset of project staff, comparing a different time span with historical data).

The third module is based upon a budgeting example. Visicalc is used to demonstrate the setup and modification of an annual budget, Apple Plot is used to graph the budget data, and CCA is used to make comparisons with historical data. The emphasis in this example is on making "what-if" projections, i.e., exploring the effects of a large number of contingencies of the budget. As before, the managers have to complete an exercise which requires them to use all three programs in the budgeting process.

The fourth module involves the CCA Database Management System. Managers are shown how to use the program to set up an electronic filing system with keyword search and cross-referencing capabilities. They are also shown how to file results from their work with Visicalc or Apple Plot. Following the demonstrations, the managers again have hands-on experience using the file system illustrated.

In the last module, other demonstrations extend the applications they have already seen. This includes variations on the project staffing chart (e.g., a project timetable, an employee work schedule), the annual budget (e.g., a sales forecast, profit/loss statement) and information retrieval (e.g., the use of an information utility to access public databases and store information of personal interest). This is followed by a group discussion on the strengths and limitations of microcomputers for managers based upon their experiences in the workshop. Finally, there is a brief discussion of the important considerations in selecting and buying a microcomputer and sources of further information (with accompanying job aids and reference reading).

## Small Business Computing

The major goal of the second workshop was similar to the one just described, but the audience and scope are quite different. The purpose of this workshop was to provide small business owners (and their employees) with an orientation to the use of microcomputers in application areas of particular interest to them. It was emphasized that the workshop would only introduce them to the potential uses of a microcomputer, not directly teach them how to use one. Recent studies have shown that a large proportion (e.g., 80 percent) of small businesses do not currently use computers and it can be anticipated that this will represent one of the most important areas for computer literacy in the 80s

The workshop focused on the following three areas: accounting, planning, and word processing. Figure 2 shows the workshop outline. The first module introduces the participants to the purpose of the workshop and the particular application areas covered. The second module introduces the hardware and software components

Figure 2

MICROCOMPUTERS FOR SMALL BUSINESS WORKSHOP

**Module 1: Introduction**
- Orientation to purpose of Seminar
- Overview of applications

**Module 2: Getting Started**
- Components of system
- How to use system

**Module 3: Basic Accounting**
- Overview of Peachtree accounting programs
- Demonstration of computer use in typical bookkeeping example
- Hands-on exercise

**Module 4: Business Planning**
- Overview of Visicalc program
- Demonstration of computer use in various planning activities
- Hands-on exercise

**Module 5: Word Processing**
- Overview of Easywriter program
- Demonstration of basic creation and editing capabilities
- Hands-on exercise

**Module 6: Wrap-Up**
- Discussion of sources of further assistance (e.g., manuals, stores, magazines, books, hotlines, etc.)

and how they are used. As in the previous workshop, we focused only on the essential operating knowledge needed for the workshop and omitted any general discussion of how the machine works, programming languages, etc.

Module 3 is an introduction to the use of the microcomputer for the basic accounting activities of general ledger, accounts receivable, and accounts payable using Peachtree accounting software. A case study was introduced and the use of the programs for each activity was demonstrated. The accounting databases were already created, so the participants could perform various updating transactions and generate reports. After the demonstration, participants were given an exercise which required them to perform some simple transactions with the programs.

In Module 4, the Visicalc program was used to illustrate the use of a microcomputer for different types of business planning. This was similar to the use of Visicalc in the "Micros for Managers" seminar except that the examples were oriented towards the interests of small business. Examples included inventory status, a balance sheet, an employee work schedule, and the comparison of leased versus purchased equipment using a net present value calculation. After each of these examples had been demonstrated, the participants had an opportunity to use Visicalc to further manipulate the examples discussed.

Module 5 introduced the participants to the use of a microcomputer for word processing with the Easywriter program. The basic steps in the use of a word processing system were demonstrated (i.e., creation, editing, formatting, printing) in the context of a simple document. Following the demonstration, participants had an opportunity to practice these steps with their own document.

In the last module, we discussed sources of further assistance available to them including the manuals and documentation accompanying the programs, store personnel, useful magazines and books, and problem "hotlines" provided by some of the software companies.

## CBT Literacy

The third workshop introduces training managers and developers to the opportunities and logistics associated with Computer-Based Training (CBT) with particular focus on microcomputers. If microcomputers are to be widely used in business and industry, it follows that training departments will ultimately have to become fully aware of their potential training uses as well as acquiring the capability to support CBT activity. A recent survey conducted by HumRRO (Kearsley, Hillelsohn & Seidel, 1981) indicates that the use of microcomputers for training is increasing dramatically.

Unlike the previous two workshops described which lasted for half a day, this workshop spanned one full day and four half days (a total of 3 days). Figure 3 shows the outline for the CBT workshop. Of the five modules shown 2, 3, and 4 were primarily intended for instructional developers while Modules 1 and 5 were primarily for training managers.

Module 1 provided an overview of the different ways that computers can be used in a training environment. This includes their use for testing, for instructional management, for drills, tutorial and socratic instruction (CAI), and for simulations and database inquiry (CAL). We used a combination of overheads, slides, videotapes, and live software demonstrations to show a range of examples of each type of CBT. In addition, participants had an opportunity to try out selected programs (on the Apples) throughout the Module. The intent of this first module was to define the concept of CBT in as

Figure 3

**CBT LITERACY WORKSHOP**

**Module 1: What is CBT ?**
- Computer Assisted Testing (CAT)
- Computer Managed Instruction (CMI)
- Computer Assisted Instruction (CAI)
- Computer Aided Learning (CAL)

**Module 2: Developing CBT Materials**
- Instructional systems design and CBT
- Authoring process and procedures
- Author language and authoring systems
- Shell games

**Module 3: Authoring CBT**
- Hands-on practice with pilot

**Module 4: Revising CBT**
- Revision of Pilot lessons

**Module 5: CBT Implementation Considerations**
- Limitations
- Enhancements
- System alternatives
- Cost/Benefits

broad a fashion as possible using examples to illustrate the concepts.

The second module focused on the knowledge and skills involved in the actual development of CBT materials. The module started with a discussion of the importance of using good instructional design principles in the development of CBT curriculum and was illustrated by CBT examples which did or didn't follow these principles. This was followed by a discussion of the steps involved in developing CBT materials in comparison to other instructional media. It also included a discussion of the type of resources (particularly human) required to develop CBT. Next, author languages and authoring systems were introduced and discussed. After this discussion, the participants spent the rest of the module learning how to use a simple authoring system, Shell Games, to develop test items.

Module 3 provided hands-on practice to develop CAI lessons using an author language (PILOT). The module began with a brief introduction to the essential components of a programming language and PILOT. However, we avoided teaching any programming concepts which were not needed for the workshop such as variables, loops, conditional logic, etc. By means of demonstrations and job aids, we had the participants work through a series of problems beginning with a simple text display leading up to a complete lesson with complex keyword answer analysis. By the end of the module, the participants had learned enough PILOT to be able to write simple CAI lessons and had a first-hand feel for what the authoring process involved.

In Module 4, participants had an opportunity to try out each other's lessons and thereby generate performance data. On the basis of this performance data, they revised their lessons, making improvements in the answer analysis and remedial instruction capabilities of the programs. The module also included a general discussion of the use of performance data based upon actual examples of performance data reports from a variety of CBT systems.

The last module addressed a variety of CBT implementation considerations. This included a discussion of the limitations of CBT, enhancements (e.g., touch input, speech input/output, graphics, videodisc, information networks), system alternatives, and cost/benefits. The module involved actual demonstrations, slides and videotapes of enhancements and system alternatives.

## What We Learned

In general, each of the three workshops achieved its major goals in terms of computer literacy. However, each workshop was less successful than we might have hoped for. In the case of the "Micros for Managers" workshop, there was a problem due to the context in which it was presented. It was given as a separate, stand-alone part of a week-long management training program. While the managers could see the value of using microcomputers for the applications we demonstrated, they subsequently went through a full management program in which microcomputers were not used at all. This made it difficult for them to realize that microcomputers could be used for may of the things that they were learning in the rest of the program. We concluded that rather than provide a separate microcomputer workshop, the management training program should be revised to include microcomputer applications wherever appropriate. This way, the use of a microcomputer could be properly seen as a standard management tool rather than as an isolated or special activity.

In the small business workshop, we encountered a problem which had been anticipated in advance but couldn't do anything about: the diversity of interests and backgrounds. For example, an accountant and an accounting clerk (both from small businesses) have quite different needs and questions when it comes to a basic accounting system. Similarly, the owner of a small manufacturing company and a clerk-typist in that company have different interest levels when it comes to planning and word processing. Furthermore, individuals from different types of small businesses have different concerns. For example, the bookkeeping in a retail business (such as a grocery or clothing store) is focused on accounts payable (since almost all receivables are cash), whereas in a service business (such as a dentist or construction company), accounts receivable are of most importance. The obvious solution to this dilemma is workshops which deal with a single application area and hence are able to better deal with the diversity of interests. The ultimate solution, of course, is individualized instruction (e.g., CAI) in which each participant is taught at exactly their level of interest.

The major problem in the CBT Workshop was caused by our attempt to meet the needs of both training managers and developers in the same workshop. Managers were generally interested in knowing about things while the developers wanted to do things. Thus, while both groups found the content of the first and last modules relevant and important, they had different needs in terms of its delivery. Therefore we concluded that the existing workshop should be split into two completely separate workshops for managers and developers.

In addition to these specific outcomes, we learned some general concepts about microcomputer-based workshops. We learned that the best strategy for teaching the use of the computer involved a lecture/demonstration/hands-on cycle in which the basic function or capability to be learned is explained, followed by a demonstration, followed by hands-on practice which either mimics or closely resembles the demonstration. This three part cycle is repeated for each computer function or capability being taught.

The classroom set-up we evolved for this approach involved the use of overheads for the lecture, then slaving each participant's computer monitor to the instructor's machine for the demonstration, followed by hooking the participants' monitors back to their machines for the hands-on. We also used two large screen (19 inch) TV monitors slaved to the instructor's machine to point out things during the demonstrations. In the CBT Workshop, these monitors were also needed for the videotapes. In order to continue with a new concept or topic after a hands-on session, the participants' machines were slaved again to the instructor's terminal (although this frequently brought cries of outrage from participants who wanted to continue with what they were doing).

The other major concept we learned from these workshops was that of system transparency, i.e., minimizing the amount of attention paid to the details of the hardware or software and focusing on the application. Thus, we taught only the bare essentials needed to use the microcomputers and programs and spent almost all of the time teaching what the system could do. While it is probably true that to become a sophisticated microcomputer user, it is necessary to understand about hardware and software, we do not feel that this is the way to introduce new users. Instead, we recommend showing new users what a particular program can do for them and getting them to try it as soon as possible.

## Conclusions

On the basis of our experiences with these workshops, we conclude that computer literacy activities for business should have the following characteristics:

- the applications selected for demonstration should be directly applicable to the employee's job
- A combination of demonstration and example is needed to provide a sense of comfort and control
- hands-on exercises should be planned carefully in advance, with simple databases or examples provided on disk
- applications should be based upon widely available off-the-shelf software to minimize the barriers to use on the job
- participants in computer literacy seminars should be as homogeneous as possible in terms of job responsibilities/interests

Probably the most important principle of computer literacy training is that computer use should be taught in the context of actual job tasks (functional literacy). The computer is a tool, not an end in itself. Thus, computer applications should be integrated into the mainstream of regular job-related training rather than isolated as separate courses (as we did in our workshops).

A second general principle is the need to customize computer literacy activities to a particular organization. In a sense, organizations are like individuals in terms of having their own idiosyncrasies and ways of doing things. Even though two companies may be in essentially the same business, they may have quite different management philosophies, operational procedures, or job functions. For this reason it is important to tailor the content and instruction of a computer literacy program to suit the needs of the organization.

The third general principle is that different levels of employees have different needs with respect to computer literacy. This was clearest in our CBT literacy seminar where managers wanted to know about things whereas developers wanted to learn to do things. This difference is also true of accountants or comptrollers versus accounting clerks and first line supervisors versus middle or senior managers. To properly address these different levels of computer literacy needs it is important to have the instruction focused on a single level and a homogeneous group. To the extent that computer literacy training is integrated into job training, this will be essentially taken care of.

The driving force underlying computer programs in business and industry is the desire to increase workforce productivity. Having "computer literate" employees is likely to be a very critical factor in the ability of an organization to stay competitive in our increasingly technological society. In fact, it is not inappropriate to view the ability to understand and interact comfortably with computer technology as a basic skill which employers will require of their employees. While public education is now rising to the challenge of providing universal computer literacy, for the near future it will be necessary for business and industry to look after its own computer literacy needs.

## Note

Apple II Plus, Apple Plot, and Shell Games are trademarks of Apple Computer, Inc. Visicalc and CCA DBMS are trademarks of Personal Software, Inc. Peachtree is a trademark of Peachtree Software, Inc. Easywriter is a trademark of IUS, Inc. Mention of these products is not to be considered an endorsement by the authors.

I would like to acknowledge the collaboration of Michael Hillelsohn and Beverly Hunter in this work.

## References

Clogston, T. CBI for a high-tech industry. Instructional Innovation. September 1980, 22-24

(References Cont'd)

Hunter, B. Computer literacy in grades K-8. Journal of Educational Technology Systems, 10(1), 1981-82, 59-66.

Kearsley, G., Hillelsohn, M.J., & Seidel, R.J. Microcomputer-based training in business and industry: Present status and future prospects. Journal of Educational Technology Systems, 10(2) 1981-82, 101-108.

Seaman, J. Microcomputers invade the executive suite. Computer Decisions, February, 1981.

Seidel, R.J., Anderson, R., & Hunter, B. Computer Literacy: Issues and Directions for 1985. New York: Academic Press, 1981.

# ON SOFTWARE TRAINING VIA CAI:

# SOME PROBLEMS AND POSSIBLE SOLUTIONS

Michael J. Zimmerman

Digital Equipment Corporation

## ABSTRACT

In developing Computer Aided Instruction, realism has generally been traded off against cost. The emphasis toward making the user interface as realistic as possible generally results in an increase in programming time and expense.

In CAI for computer software, this realism primarily centers around the creation of CAI simulations of the computer software. The essential problem is that in order to make the simulation appear as much as possible like the software being studied, the programmer has to create CAI software that approaches the simulated software in its complexity.

This paper and the accompanying film, discusses an alternative approach using windowing and subprocesses on the VAX 11/780 which allows the student to actually use, in a controlled environment, the software that is being studied.

The CAI maintains the control and yet provides a softening impact on the students' first encounter with the real software.

## INTRODUCTION

One of the suggested topics for this year's ADCIS conference underscores research being done by Digital Equipment Corporation's Computer-Based Course Development group (CBCD).

> "Use of computers in instruction has attempted to adapt the instructional process to the individual student in the most friendly, encouraging and effective ways possible; however, programming the necessary interactions has been extremely difficult. The level of adaptivity has therefore been a tradeoff with courseware development expense."

To minimize the need for this trade off, CBCD has adopted an approach which saves many hours in program development and provides a higher quality of courseware at the same time. The basis of this approach is to bring the software being taught into the instruction in a restrictive setting.

The effect is not unlike having a tutor watching over your shoulder, and commenting when needed while you use the actual software. Cost savings are acheived by avoiding the complicated simulation that normally accompanies any but the most simplistic effort. Although Digital develops CBI for various computer software, and the audience is primarily semi-computer-literate, this technique is not restricted to a software setting.

The power of this technique is best understood from an historical perspective.

## THE DEVELOPMENT OF EDTCAI

Early in the summer of 1980, CBCD produced EDTCAI, an introductory, computer-assisted instruction, (CAI) course that teaches how to use EDT, the DEC standard editor. The course serves to orient the users the basic features of EDT, including line mode, where text is entered line by line, and the newer keypad mode in which users see the document on their terminal screens and can modify it in much the same way that one would modify a paper document, e.g., deleting words, sentences, inserting new lines, and so forth. EDTCAI not only provides an explanation of the editing function, but, in the context of the exercises, allows users to "try" certain commands and then to watch their results as a simulated editor moves lines, deletes words, or does whatever the particular command requires.

We encountered several problems when analyzing students' input for the line mode exercises:

- First, there were usually a variety of abbreviations for the basic command itself.
- Second, there was a wide range of possible syntaxes for referencing line numbers.

The solution to these types of problems is generally to have a programmer construct a "parser" which would check student input against a table of acceptable values. This is, in fact, just what the editor does itself. One problem with this approach is that there is always another alternate variation that was overlooked and omitted from the table. In addition, as the EDT editor itself expanded in later versions, these syntax rules broadened and, in some cases, changed.

To maintain accuracy, the parsing table used by the CAI simulation began to approach the size of the parsing table used in the editor itself. Our response to this problem was to make a "model" of the editor. Nevertheless, the benefits of a graphic simulation resulting from an acceptable as well as an incorrect answer provided a strong conceptual basis for understanding the editor.

The difficulties in parsing line mode commands were further compounded when we attempted to simulate the visual keypad editor, where each key pressed resulted in an editing action. Moving from the one dimensionality of line mode commands to the multi-dimensionality of keypad screen commands resulted in a dramatic multiplying of the parsing problem, because moving from one point on the screen to another can literally be accomplished in an infinite number of ways. Using the keypad keys to move forward and backward by words and lines provided a significant exercise in logistics. Namely, we had to keep a road map of what the text looked like and constantly calculate the position, because the action to be taken depended upon where one was when the keypad key was pressed. If it altered the text itself in any way, we had to revise and update the road map. This sort of programming begins to approach the functionality of the editor itself and indeed the editor is far better at doing that sort of thing than our CAI courses.

## MAKING THE EDITOR PART OF THE CAI

When you begin to consider the multitude of features in the real editor, subtle changes with new versions, and our programmers' labor in creating parsing tables and programming models, it was not a large step to try to use the EDT editor to "simulate" itself in keypad mode. The simulation centers around suitably restricting the functionality of the keypad editor keys so that dangerous and undesired keys have little or no effect. With one fell swoop we hoped to furlough our programmers, synchronize editor version updates (the editor is always itself), and immunize ourselves against programming errors.

This jump was not without its consequences. In the real editor, there is little opportunity for feedback to the student other than directly showing the consequences of the editing action. If the student destroys the text beyond recognition, there must be ways to get help and begin again or return to the CAI instruction for more coaching. In addition, there is little way to know whether or not the student was successful, much less where and what problems arose with his or her editing sessions. Passing such information between the editor and the CAI programs is problematic due to the independent nature of these programs.

A CAI course and an editor are both "images" which run in a "process". A process can only execute one image at a time. Images can be sequentially stacked, but when VAX/VMS switches images (as it must in going from the course to the editor), it necessarily loses "image" context. We can, however, keep track of where in the course a student is working and store the data in a file before moving into the editor. However, once the student is in the editor, there is limited ability to give feedback and branch, which is what CAI does best. Students are, so to speak, on their own.

One solution would be to control the visual output on the terminal so that the "context" remains at least in part on the screen, and the editor is restricted to another part. This is called "windowing", and is still on our wish list. It places special and specific demands on being able to do two separate things and merge the output, and the implications of this are enormous.

The Computer Science Department at Brown University had done extensive work with windowing on a VAX 11/780 running under the UNIX operating system. (The same software can be made to run on the VAX/VMS system with minor modifications.) A video tape shown at the Conference demonstrates a VAX/UNIX system. This video tape is the result of work done by the Computer Science Department at Brown University.

At present, we can sequentially execute different images by switching between a program such as the system editor and the CAI course about that program. This is done by temporarily storing the context relevant data from the CAI course before leaving the course. This has another dividend: it could be employed to place a user into a CAI course at a specific point based on the desired content. Thus, in a manner similar to on-line Help, CAI might be made randomly accessible for the specific instruction on a specific topic. If this instruction was presented in a subprocess, the original process context would not be lost. These capabilties exist now and are in prototype design. The central focus for growth now seems to lie with windowing.

Brown University has been able to window on any type of terminal under their VAX/UNIX system, including the popular VT100. (The results of their work have been published in "BRUWIN: An Adaptable Design Startegy for Window Manager/ Virtual Terminal System" by Norman Meyrowitz and Margaret Mozer, available from Brown University.) This work is in a general setting, but its consequences are clear: multiprocess windowed output is not only possible, but it is highly desirable for CAI courseware.

## FUTURE CONSIDERATIONS

As students make the transition from a controlled CAI environment to an actual software utility, there is a jump between the simulation and the "real thing". Among other things, learning attitudes change. While users may make this jump only once for each subject or software utility that they learn, it is nevertheless psychologically very significant. We are primarily concerned with easing the problems caused by this transition. As the distinction between on-line documentation and on-line training becomes more obscure, the gap that lies between CAI and the software it is teaching will disappear. As this occurs, interactive training will become an integral part of the user's entire on-line environment.

# YEAR 7 - FEDERAL AVIATION ADMINISTRATION

## CBT STATUS REPORT

JOHN A. BUCK

FEDERAL AVIATION ADMINISTRATION

### ABSTRACT

The FAA has been developing its CBT system on a cost effective basis for seven years. Currently a 70-terminal system is delivering several CBT courses at 40 locations. This system may be expanded to more than 200 locations. CBT growth entails significant training program changes.

### BACKGROUND

The Federal Aviation Administration (FAA) has been exploring and using computer-based training (CBT) to deliver various kinds of technical training since 1974. A test of a large scale system began in November 1981. If that test system continues to operate successfully, the FAA plans to install a larger, more permanent full system in the next year or two -- hopefully in time to mark the CBT program's 10th anniversary. This article summarizes some of the FAA's accomplishments to date and some of the challenges its CBT program still faces.

The FAA employs about 45,000 persons most of whom work in the air traffic control, electronics maintenance technician, and aviation safety occupations at several hundred locations throughout the country. The FAA maintains a sizeable, centrally controlled technical training system to impart the many specialized skills and knowledges required by those occupations. Most of this technical training is now provided at the FAA Academy in Oklahoma City, Oklahoma. The FAA Academy employs about 600 instructors who are drawn from the Agency's technical workforce.

The initial work in CBT concentrated on establishing that FAA students could learn effectively through a computer. Various computer-assisted instruction (CAI) 1/ lessons were developed and successfully replaced portions of classroom instruction in several courses. In one particularly notable instance, a simulation of a complex piece of equipment used on certain specialized FAA aircraft saved $70,000 net annually in equipment and personnel costs.

Later, when vendor supplied computer managed instruction (CMI) software became available, an entire two-week classroom course was converted to a self-paced CBT course. The course included CAI, textbooks, and audiovisual learning resources. CBT terminals were placed in several working FAA field facilities around the country. As is the case with all but the largest FAA installations, these facilities did not have a professional instructor staff. Students took the course at these field terminals rather than traveling to the FAA Academy to take it. Time required to complete the course was reduced by at least one-third. The experiment was deemed successful because all students were able to complete the regular follow-on course without difficulty.

---

1/ In the FAA's parlance CBT is a composite of CAI and computer-managed instruction (CMI), that is, CBT = CAI + CMI. In CAI the computer actually delivers instruction; in CMI the computer manages students' progress, gives tests, and keeps records but does not require computerized delivery of instruction.

Primarily because of these initial efforts the FAA no longer questions whether CBT "works," that is, whether employees can effectively learn certain kinds of job-related skills and knowledges via CBT in a formal school setting. However, it is still in the process of establishing the extent to which CBT courses can be successfully delivered to students at small, geographically scattered worksites where no instructor or professional training support staff is physically present.

## COST BENEFIT

FAA's current direction with CBT is based on the conclusions of a comprehensive feasibility study completed in August 1979. 2/ The study identified 40 courses with high student population and compared several computerized and noncomputerized alternatives for delivering them. It concluded that a geographically distributed CBT system was the most desirable option from the standpoint of "hard" cost savings as well as the less quantifiable benefits such as improved training standardization and scheduling flexibility. An acceptable "fallback" alternative would be a CBT system located only at the FAA Academy. Factors included in the analysis model were FAA Academy instructor salaries, student travel and per diem, training equipment and telecommunications, and contractual course development costs. Student salary savings were not considered as hard savings, at least not in the near term. Had they been included, estimated savings would have approximately doubled.

The study indicated that the FAA could expect a "hard money" payback on initial investment of about 5-to-1, assuming a 7-year system life cycle, and an annual saving of about four million dollars. A risk analysis showed that these conclusions were relatively insensitive to a wide range of variance in all costing assumptions. Confidence in the study was so high that the FAA's training organization voluntarily reduced its Fiscal Year 1982 budget by several hundred thousand dollars, specifically in expectation of the initial savings from the system.

Although cost payback is obviously an important consideration, the study observed that the biggest benefit of CBT may be the less quantifiable advantages, including better training standardization, easily available refresher and update training, more flexible student scheduling, more rigorous course development, improved training performance records, and better student retention of course information. An important potential disadvantage of distributed CBT is the possibility that undesirable amounts of time will be required by local facility management to handle the students who now will be studying in their offices rather than in a classroom at the FAA Academy in Oklahoma City.

To reduce this not so quantifiable cost of the new way of doing business, the FAA is devoting a great deal of attention to developing an administrative system in which a geographically remote course manager/instructor located at the FAA Academy manages students' progress through a CBT terminal and telephone calls as needed. For now, the FAA is strongly encouraging supervisors to put their employees in full time training status once they have enrolled in a course to minimize day-to-day scheduling difficulties. The near term experience will show how successful these and other measures are in creating a practical system on a moderately large scale.

## IMPLEMENTATION

The feasibility study recommended implementation in phases. A stage dubbed the "Interim System" is now in place. The Interim System includes almost 40 leased PLATO(R) terminals located one each at FAA operational facilities throughout the United States. About thirty-five more terminals are located at the FAA Academy in Oklahoma City. Because special housing is available it is normally cheapest to send students who do not work near a field terminal to Oklahoma City.

Nine courses with about 1000 students annually are being converted to CBT from conventional (that is classroom) instruction for the Interim System. The

---

2/a "Requirements Analysis and Feasibility Study of Computer-Based Instruction for FAA," Buck, Bullard, & Fagan; ADCIS Proceedings 1979, Vol 2, pp 331-338.

2/b "How Computer-Based Instruction Was Cost Justified In the FAA, "Buck, Beardsley; ADCIS Proceedings 1980.

courses range from elementary courses on electronics theory to moderately advanced pilot ground schools and equipment courses in which certain "hands on" skills are taught through sophisticated interactive simulations of actual equipment.

Students study at completely equipped training centers located on the premises of their worksite. A training center consists of four individual carrels and a worktable. Each carrel is devoted to a particular kind of learning resource such as a videotape monitor, a videoslide player, an empty carrel for reading workbooks and manuals, and a CBT terminal. The worktable provides space for students to work on electronic "hands on training labs".

Students move through their courses individually, guided to the different learning resources and tested at the end of each module by the computer. Normally no more than 10% of a student's actual learning time is in CAI. The rest of the learning time is spent with the other learning resources. If a student has a question or gets stuck, he/she may exchange notes over the system with the instructor or call on the phone.

## FULL SYSTEM

The "CBT Full System" is to follow the Interim System. Once the FAA has gained enough experience with the Interim System to be sure of its training administration requirements, it plans to buy equipment with lower operating costs. (Currently it is leasing terminals, timeshare services and phone lines.) It expects to increase the number of CBT courses and expand coverage up to 200 facility locations throughout the United States and to Japan and Germany.

The Full System equipment will support course material written in more than one CBT authoring language. In addition to giving access to the broadest possible range of developed courseware, this arrangement should help assure flexibility, enabling FAA to take advantage of such anticipated improvements in software technology as adaptive testing and artificial intelligence. Also planned is the flexibility to incorporate maturing new hardware such as videodisc.

## CONCLUSION

The FAA has established to its satisfaction that computer-based training (CBT) should replace a substantial portion of its current classroom training. Properly used, CBT can significantly improve training efficiency and effectiveness. Current FAA planning envisions methodical expansion of the present system of 75 terminals to serve more than 200 locations, conversion of more classroom training, and development of most new training in CBT format rather than traditional modes. The change to CBT is entailing significant changes in the way the FAA training program is administered throughout the agency.

# PROJECT U-TRAIN CAL

## AUTHORING METHODS FOR SUBJECT MATTER EXPERTS USING NATAL-II

Lt(N) Griff Richards

Major Jean Bélanger

### ABSTRACT

Project U-TRAIN/CAL is an investigation by the Canadian Forces into the implications of Computer Assisted Learning for self-paced technical training. One central question has been the traditional role of the subject matter expert (SME) as the technical instructor in military training. This paper discusses training and support of SME author/programmers and discusses the preliminary results of their courseware developments using NATALL-II.

### INTRODUCTION

Project U-Train/CAL is a five year research and development project of the Canadian Forces into the implications of Computer Assisted Learning (CAL) for self-paced technical training. The project is conducted by the Training Development Division of the CF Training System Headquarters with support of the Chief, Research and Development, the National Research Council of Canada and 426 Transport Training Squadron.

While the project has many far-reaching objectives including the test of NRC's NATAL II CAL system on the Honeywell Level 6 machine and evaluation of multi-media terminal configurations, it is inevitably the manpower aspects which will make or break any attempt to move into the large scale development of CAL courseware. Two questions are central here: Can Canadian Forces technical instructors prepare CAL courseware, and if so, what assistance and support will they require?

### AUTHOR SELECTION AND TRAINING

When U-Train/CAL commenced in 1979, several classroom technical instructors were selected for part-time participation as author-programmers. The selection was made on the basis of subject-matter expertise plus a minimum of one year instructional experience in the course to be developed for CAL.

Authors attended a short course on the creation of self-paced materials, and were given an introduction to CAL programming using NATAL-74 (Bélanger, Madsen and Drake, 1980). From that point onwards, the authors followed the instructional systems design (ISD) model (Figure 1) on a time-available basis. As each draft of the Task Analysis or Content Scripting was finished it was checked by the project's Training Development Officer and sent for independent review and technical check by other subject matter experts (SMEs).

Progress on each course varied depending on how much time was available to the authors. Although at times authors would be away from the courseware development for over a month, their adherence to a common model made it possible to start up again after each delay with minimal disruption. At time of writing (Dec 81) no course has yet reached the evaluation stage. The following discusses some of the tactics used for courseware development.

### ANALYSIS

Analysis was performed at the managerial and task levels. The former was conducted to determine which courses would benefit most from CAL in terms of type of training and resource justification. Once a course had been selected, the assigned authoring team conducted a task analysis to determine need-to-know objectives of the course. Figure 2 shows one page of a completed task analysis which follows a format developed by Friesen, Kaye and Associates of Ottawa and resolves tasks down to a level at which criteria questions are developed. The task analysis is reviewed by other subject-matter experts, supervisors of potential course graduates and, if feasible, the original equipment manufacturer. When a consensus has been reached and the final draft has been accepted, development can move on to the design phase.

## DESIGN

Using the task analysis as a blue print, the author prepared a "script" of the course content required to meet the criteria questions. Figure 3 shows one page of a content script prepared in a loose-leaf binder. The paper format allows for insertion of graphics and questions as required, and is easy to circulate for technical review and editing. As the scripting progresses areas requiring simulation or other special development can be singled out. Once sections of the course have been scripted to the satisfaction of all concerned, production can begin.

An unexpected use of one course's content script was its adaptation by one training unit as an interim self-paced, paper-based module of instruction.

## PRODUCTION OF COURSEWARE

As each author varied in his amounts of contiguous time available for CAL, typing skills, programming expertise and desire to see the course running, the authors were given the freedom to choose their own programming system. Four of the methods are contrasted below in ascending order of effectiveness.

### 1. Extemporization

In this first approach, an author with a fair amount of programming experience, short-cut some of the content scripting and wrote material directly from the task analysis into NATAL units - i.e. creating both content and format simultaneously. While this reduced the initial workload of preparing a content script, substantial amounts of code had to be re-authored as the errors normally caught in a script review now appeared in executable code. It is estimated that the author spent an extra 15 per cent more time over any of the other methods due to the amount of backtracking and re-working of NATAL code.

### 2. Enter and Edit

The second method, used by a poor typist, consisted of having a data entry operator type the content script into the computer system. The author then partitioned and formatted the material into teaching frames through the use of the on-line text editor. The NATAL code was in effect created around the course content. Although this "enter and edit" provided a very rapid means of storing the material, several drawbacks were encountered:

1) framing by editing encouraged use of "AS IS" formatting for a particular terminal size and made little use of the flexibility offered by NATAL's Display Sub-Language (DSL);
2) as only the author knew how he conceptualized the frames, he himself had to do all of the on-line editing and this became very tedious;
3) in efforts to streamline the editing process, the author developed generic display units to which long string parameters would be passed for screen output. The result was inefficient (i.e. slow) program code which was difficult to follow for other programmers.

After a sufficient level of run-time and debugging frustration was reached, extra assistance was given to the author and the course was restructured into the natural flow of a NATAL program. This resulted in a program of much reduced size with easily followed code and faster execution.

### 3. Piecemeal

In another approach, an author directly coded small sections of his content script into NATAL. Display commands to format the screen were inserted as thought required - almost on a trial and error basis. The code was then entered into the system for proofreading and debugging. As each "chunk" of code frames was too small to warrant the use of a data-entry operator, the author was his own two-finger typist. While piecemeal programming proved to be very time consuming, it did result in self-documenting code modules and the author could see a progressive accumulation of executable courseware.

### 4. Format, Code and Polish

In the fourth approach, the author framed his content script onto specially prepared screen lay-out sheets (Figure 4). Special instructions for programming could be noted in margins, and any required graphics could be sketched on the back of the sheet. Once the screen designs had been reviewed and corrected the appropriate NATAL code was written onto coding sheets (Figure 5) which were passed to an operator for data entry. Once the code had been entered into the computer system, the author and additional programmers began debugging the courseware. As each section was finished it was integrated into the emerging course.

Although the design of screen format is an extra step, it is similar to a storyboard and allows for review of the design concept before programming takes place. Even after checking the content script, the field users and subject-matter experts who reviewed the screen designs were able to make valuable suggestions on the use of animations and 35mm slides. Of all comments reviewed, less than 10 per cent dealt with the technical content, while about 20 per cent dealt with grammar and style.

In short, the advantages of this method are:

1) an additional review of the design takes place;
2) coding is easier since the display format has been already determined;
3) coding can be carried out by programmers not involved with the content development;
4) the screen design serves as a permanent copy of the author's intent and run-time displays can be easily checked against the screen design when polishing the program.

## EVALUATION

Formative evaluation took place as lessons under development reached completion of one of the stages discussed above. Whenever possible, students representative of the target population were asked to follow part of a content script or completed lessons of instruction. If possible, the author would be present during the test sessions to gauge the effectiveness of his materials and ferret out any problems with the materials. Not surprisingly, the further along in development a course was, the fewer and finer were the students' criticisms. Most of the initial difficulties were encountered with ambiguous or insufficient instruction to the student as to what to do next, or which information to attend to. Second in frequency were incomplete reprise messages to respond to incorrect answers and thirdly, were general grammar and spelling errors.

While it is not possible to test each and every module during formative evaluation, it is anticipated that students' comments and automatic data collection during runs of the completed courses will enable continued improvement of the courseware. It is intended that samples of students scheduled into the traditional classroom-delivered course will be funnelled through the CAL course until sufficient reliability is established to schedule students to attend the CAL course and phase out the classroom instruction. It is anticipated that two or three years may be required before the courseware matures enough to be exported for remote coursing without the on-site availability of a qualified instructor.

## SUMMARY

The specific implementation of a general model of instructional systems design for CAL using subject-matter experts (SME) has been discussed. In general it was found that the SME instructors were very receptive to CAL and many were eager to participate in the project. Although the instructors had little difficulty in learning the principles of self-paced instruction and CAL programming using NATAL-II, it was found that assistance was required in performing task analysis, creating control structures for NATAL programs, and in the design of graphics.

Instructors found that formative evaluation of both planning materials and programmed lessons was a valuable part of CAL development. Systematic design of courseware, including the designing of screen formats allowed for excellent review of the content by subject-matter experts and the project managers.

The authors of this paper highly recommend the use of screen lay-out sheets as a mandatory step in courseware development, and discourage extemporisation of courseware. In addition, it is recommended the "natural" format of NATAL UNITS be used rather than generic UNITS (with string parameters) as the resultant code is self-documenting and easier to decipher for courseware maintenance. Coding sheets are indispensible if the services of a data entry operator are to be used.

## CONCLUSIONS

The more paper-based stages of a given piece of courseware that are produced, the faster and cleaner will be programming and debugging of the code.

CAL authors should be encouraged to use screen format sheets to design their courseware before coding. This record is faster to produce than on-line code, can be more readily reviewed and revised, and permits programmers to assist in course preparation and debugging. The use of separate coding sheets will permit faster and more accurate data entry by skilled typists.

NATAL is a powerful CAL language easily learned by non-programmers. While the absence of any clear template enhances the flexibility of the author to create what he wants, it can also be a pitfall for creating powerful "driver" routines difficult to follow or maintain by other programmers.

## ACKNOWLEDGEMENT

The authors would like to thank Dr Eugene Romaniuk of the University of

Alberta for his assistance in the development of the screen lay-out forms.

REFERENCE

(1) Current Application of Computer Assisted Learning in the Canadian Armed Forces, Capts J. Bélanger and A. Madsen, Sgt R. Drake, ACIT 80, Feb 80

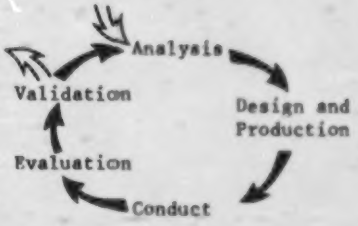
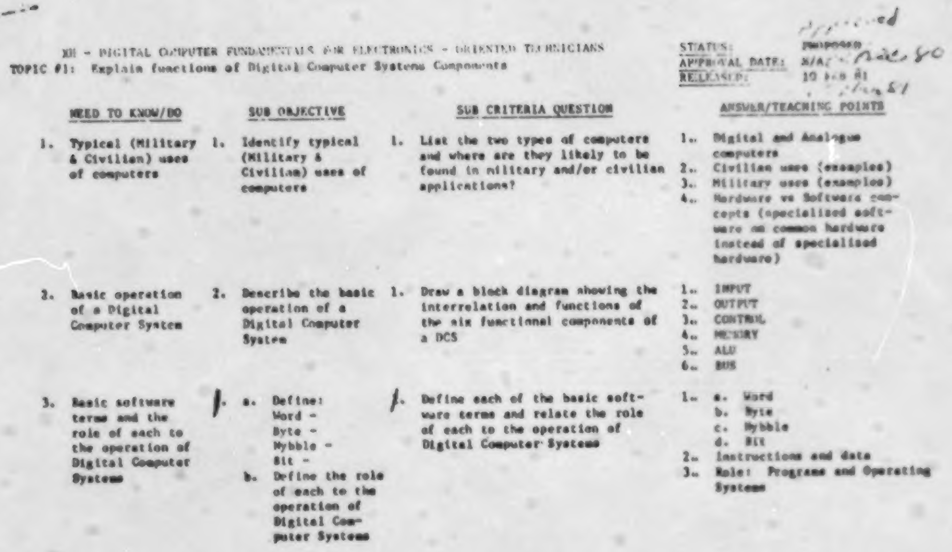


Figure 1. Instructional System Design Cycle



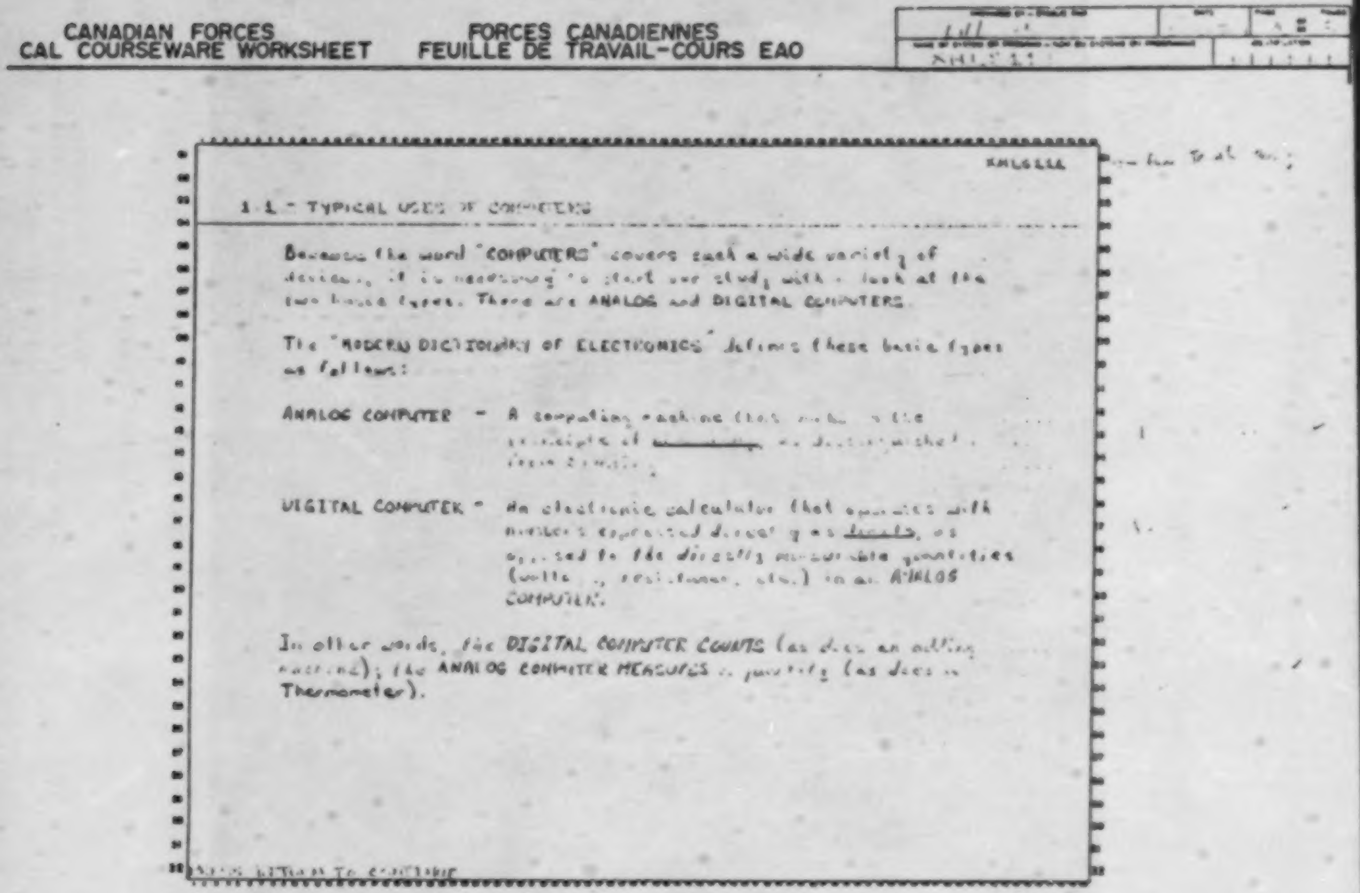
XII - DIGITAL COMPUTER FUNDAMENTALS FOR ELECTRONICS - ORIENTED TECHNICIANS
TOPIC #1: Explain functions of Digital Computer Systems Components

STATUS: ~~PROPOSED~~ Approved
APPROVAL DATE: N/A Dec 80
RELEASED: 10 Feb 81 Jan 81

| NEED TO KNOW/DO | SUB OBJECTIVE | SUB CRITERIA QUESTION | ANSWER/TEACHING POINTS |
|---|---|---|---|
| 1. Typical (Military & Civilian) uses of computers | 1. Identify typical (Military & Civilian) uses of computers | 1. List the two types of computers and where are they likely to be found in military and/or civilian applications? | 1. Digital and Analogue computers<br>2. Civilian uses (examples)<br>3. Military uses (examples)<br>4. Hardware vs Software concepts (specialized software on common hardware instead of specialized hardware) |
| 2. Basic operation of a Digital Computer System | 2. Describe the basic operation of a Digital Computer System | 1. Draw a block diagram showing the interrelation and functions of the six functional components of a DCS | 1. INPUT<br>2. OUTPUT<br>3. CONTROL<br>4. MEMORY<br>5. ALU<br>6. BUS |
| 3. Basic software terms and the role of each to the operation of Digital Computer Systems | 1. a. Define:<br>Word -<br>Byte -<br>Nybble -<br>Bit -<br>b. Define the role of each to the operation of Digital Computer Systems | 1. Define each of the basic software terms and relate the role of each to the operation of Digital Computer Systems | 1. a. Word<br>b. Byte<br>c. Nybble<br>d. Bit<br>2. Instructions and data<br>3. Role: Programs and Operating Systems |

.../2

Figure 2 Task Analysis Format



CANADIAN FORCES
CAL COURSEWARE WORKSHEET

FORCES CANADIENNES
FEUILLE DE TRAVAIL-COURS EAO

XHLG11A

1.1 - TYPICAL USES OF COMPUTERS

Because the word "COMPUTERS" covers such a wide variety of devices, it is necessary to start our study with a look at the two basic types. These are ANALOG and DIGITAL COMPUTERS.

The "MODERN DICTIONARY OF ELECTRONICS" defines these basic types as follows:

ANALOG COMPUTER - A computing machine that works on the principle of analogy, as distinguished from [illegible].

DIGITAL COMPUTER - An electronic calculator that operates with numbers expressed directly as digits, as opposed to the directly measurable quantities (voltage, resistance, etc.) in an ANALOG COMPUTER.

In other words, the DIGITAL COMPUTER COUNTS (as does an adding machine); the ANALOG COMPUTER MEASURES a quantity (as does a Thermometer).

PRESS RETURN TO CONTINUE

Figure 4 Screen layout sheet

# COMPUTER TRAINING

## FROM SECRETARIAL TO TECHNICAL PERSONNEL

JOHN GREEN - IBM CANADA LTD.

ENID SQUIRE - IBM CANADA LTD.

## ABSTRACT

Within IBM there is a requirement to offer computer training ranging from "What is it?" (computer literacy), to full technical competence in a manner that can reflect changing trends in the use of computers and the technology used to build them.

A variety of training methods is used. This paper describes two different approaches for two very different groups of people. Both methods provide the flexibility to reflect change with a minimum of effort.

Finally an indication of what the future may hold, for example, incorporating the Personal Computer, and world wide CBT.

## INTRODUCTION

Within IBM every medium is used for training and a wide range of skills is offered from secretarial skills to specialised technical courses. The two areas to be described here represent two extremes in training related to the computer. In both cases there is a requirement to keep the material current, with a minimum of effort. In some cases course material must immediately be updated, for example, when a technology becomes obsolete, the course must be changed to reflect the new material. The following topics describe two different training methods and media.

## CBT FOR CUSTOMER ENGINEERING

### The Audience

- Customer Engineers across North America. These are the people that maintain hardware and software products.

### The Requirement

- Product support training in hardware and software for new products.
- On-going training for all products.

### THE PAST

### A Feasible Study

In 1965 the IBM Corporation reviewed its method of training its customer engineering staff and arrived at the following conclusions:

- Costs were too high - living/travel/tuition.
- Costs were increasing - living/travel/salaries.
- Too much time spent away from home/territory.
- Customer dissatisfaction with CE not being available.

As a result we looked for a better way of training and maintaining the skill levels of our servicing staff.

Computer Based Training (CBT), although then relatively crude by today's standards, was chosen as the vehicle to provide part of the solution.

Some of the more obvious benefits seen from this approach were:

- Reduction in costs - facilities/travel/personnel.
- Control over costs.
- Reduction in time spent away from home/territory.
- Improved customer service.

Crude Beginnings

CBT equipment available to the field force in 1965 were hard copy terminals which were slow and noisy. CPU's could support relatively few users. There was a lack of appropriate courseware and of any experienced CBT staff.

The Language

The original courses were written in Coursewriter. Productivity was improved by the development of authoring macros and a course framework. Based on experience these macros were redesigned and in fact were marketed to our customers as ITS (Interactive Training System).

The Network

In 1968 there were 22 terminals across Canada. By June 1965 we were up to 28 terminals in 18 locations. The terminals fed into concentrator sites across the USA. These concentrators were supported by a S/360 Model 65 in Sterling Forest, New York.

By this time ten percent of CE training was done by the CBT network.

The Feedback

On the negative side:

- Response was slow.
- The terminals were noisy.

On the positive side:

- Course content was good.
- Reduced travel and inconvenience.
- Training costs decreased.

The Present

The Network

Sterling Forest
2 x 168

White Plains
Gaithersburg
Chicago
Palo Alto
Toronto

Sidney

Terminals in all major cities of the USA and Canada.

Multi-Media

Video Disks
Audio
Film Strips
Manuals
Access to real system

Statistics

There are now 400 video display terminals dedicated to this system and many more that can use the system on a part-time basis. There are printers available near the terminals and recently video disks have been added. There are approximately 500 courses available ranging in length from 1 to 90 hours. Canada's involvement is about one-tenth that of the US. No course development is done in Canada.

Course Production

Authors are usually located in manufacturing plants and work closely with the research personnel. Authors are experienced customer engineers or program support representatives. They are trained in CAI techniques, education technology, learning theories and writing techniques.

Worksheets permit translating educational plans into pre-structured program logic without computer programming expertise. A pre-test may be given to discover the level of the student. The author to student time ratio is an average of 100 to 1.

The system provides a comprehensive student record system and for each CE the system determines what courses are needed by matching the student's history with his requirements.

Why CBT?

Without the use of CBT the education of CE's would have been very difficult to accomplish in an effective manner. The savings are estimated at $4 million. This does not include living costs which at $45 per day would add $4.5 million to the cost of education.

## COMPUTER LITERACY FOR ADMINISTRATION PERSONNEL

The Audience

- Secretaries, administration personnel.

The Requirement

- Computer literacy.
- React to any questions about computing.

Introduction

Although IBM is a computer manufacturer, there are many people employed by the company who are not Data Processing specialists. It is not necessary for administrative personnel, engineers and real estate people to understand how a computer works and in fact can do their jobs perfectly well without this knowledge. However, a good secretary can become more efficient if she has some understanding of the work she is processing.

For a number of years IBM has offered a three-day course for Administration and Secretarial personnel called "Computer Concepts for Administration". The agenda of the first day includes 'How the computer works' and 'Writing a program'. The other two days are devoted to some of the ways in which IBM uses the computer, as well as visits to computer installations and customer support areas.

This paper is concerned with the first day, since the material taught is applicable to any company wishing to offer a concepts course. The course is usually offered six times per year, with 18-20 students per course. When a survey was done to find out how many people wished to enroll, it was found that 22 courses were required that year to accomodate the requests. You might say it is a popular course!

The Morning Session

- General discussion about a computer - where we see it used - where it fits - its general characteristics - will it take over our lives?
- Outline of a simple problem solved by the brain:
    - The problem - to add 2 numbers.
    - Input - 2 numbers.
    - Processing - adding the numbers using instructions stored in our brain.
    - Output - speaking or writing the result.
- Relate to a simple computer.
    - Input unit - card reader.
    - Input - punched cards.
    - Central processing unit (CPU) - stores and executes instructions.
    - Output unit - printer.
    - Output - printed paper.

```
Card     C        Printer
Reader    P
           U
```

- Explain:
    - Card code - hand out cards.
    - "Reading" by a card reader.
    - Bits, bytes and words using core storage (the easiest medium to explain).
    - Transmission from one unit to another is electronic and a change of code is effected (no further explanation here).
    - Instructions form a program that does a job - machine language.
    - Cycle for solving a problem, using a simple computer and machine language.
- Key the instructions (program) into cards.
- Key the data into cards.
- Feed the instructions through the card reader.
- Execute the instructions - will read in data when required.
- See the results on the printer.

Since machine language would be a series of ones and zeros, i.e. 11100101 11010100, there would be a problem finding errors as well as the initial problems connected with writing any meaningful program. Obviously, there was a requirement for a language nearer to the language people understand. Thus we have low level languages, using a for add, b for branch, c for compare, h for half, etc. The machine only understands machine language so a translator is required. The translator (this term is used so as to avoid confusion), is usually written by the manufacturer for this machine. A machine language is unique to a machine or series of machines.

Now let's solve the problem again using a low level language·

- Write the program using a low level language - punch into cards.
- Feed in the translator.
- Feed in the program.
- The translated program is punched to cards - explain a card punch.
- If errors - machine language is discarded.
- Fix errors and translate again.
- When all errors are fixed, run machine language cards followed by data.

Obviously there is a lot of waste of cards and the process is not "people productive". Thus we have high level languages. However, the translator for a high level language would be much larger and feeding it in via cards is out of the question. Now we add tape and disk to our simple system, with appropriate descriptions, pictures, etc.

The rest of the morning is spent expanding the problem-solving process on a computer. The main storage technology is explained (this is one explanation that has changed over the years). The problem is expanded to include files. Operating systems are discussed. The system now begins to expand to today's complexity, with the introduction of virtual storage, multi-processing, multi-programming, teleprocessing network, and mass storage. Whatever is significant today is discussed, for example, data base, distributed processing, micros vs central computing.

All of these items are illustrated with foils and pictures when possible. Extensive use is made of the blackboard. No fancy slides or foils are used since they could be obsoleted before the next course. Short films are used if available, for example, there is a good film on virtual storage called "A Much Better Way".

The questions range far and wide based on the students' backgrounds and where they work within the company. Everything from "What is a benchmark" to "How does the computer know where to start the entries on my bank book". Looks of enlightenment can be seen frequently on the students' faces as they understand some term or process used in their area.

## The Afternoon Session

The students learn sufficient APL to code and test a program using either terminals or 5100 computers. (We shall probably switch to the Personal Computer in the near future).

The history of APL is covered. Flowcharting is used to illustrate the logic of program using payroll as an example. A problem is then shown to the students in the format received by programmers, i.e. a description NOT a logical flowchart. The students are taken through the problem-solving process from beginning to end.

- Understand what you are given and what you are required to produce.
- Draw a flowchart.
- Write the program in appropriate language.
- Enter the program.
- Test and debug the program.

The students then solve their own problems working in pairs. Their progress is monitored and help given if required. They enter their programs and test them, making all the usual mistakes - typing errors, getting in a loop, one answer instead of several and so on.

## The Student Problem

- Produce a table to show the amount of money accrued at each year-end for a sum of money, invested at a fixed interest rate, compounded annually.

There are some slight differences between the problem used as an example and the one the students solve themselves.

The afternoon session finishes with a film showing computer applications. During the film they see both a flowchart and some APL code and a variety of applications worldwide.

Why Classroom?

Why not CBT? One of the opening remarks on the course is; "Ask any questions you like -- there is no such thing as a dumb question on this course". Therefore this excludes any medium other than classroom. That is not to say that an introductory course using CBT, video tape or a text book cannot be a pre-requisite. When you give the students the opportunity to ask ANYTHING that has puzzled them, you need a human being on the spot to answer the questions.

The comments received from the students about the course are very positive and to-date not one student has fallen asleep!

## An Alternative Approach

An approach tried for a group of managers at a half-day session, was to take a newspaper announcement of a new product and explain all the terms used. The session was held in an amphitheatre. Transparencies were used to provide diagrams as the lecturer talked.

## EDUCATION IN THE FUTURE

The use of the Personal Computer:

- For computer literacy training.
- For CBT
  - Stand-alone
  - As a terminal
- For pre-requisites for the course.

Training distributed worldwide from a central location:

- Advantages:
  - One location for updating.
  - Use of the computer during the day in one location and at night in another.
- Disadvantages:
  - Language differences in different countries.
  - Network considerations.
  - Getting assistance with a problem.

# HAND-HELD COMPUTERS -- FOR CBI?

Dr. Larry Francis

Battelle Memorial Institute

## ABSTRACT

Miniaturization has reduced electronic components small enough that hand-held computers with significant capabilities are now available. This paper examines the feasibility of using such devices for computer based instruction delivery. It concludes that several new uses are possible with handhelds, but that their limited display area is a major constraint.

## BACKGROUND

In the past, CBI hardware has sometimes set trends, but more often it has followed trends. Perhaps the most trend-setting CBI hardware was the touch panel built for the PLATO project. Though touch panels had been built before, their use in so many ways on PLATO demonstrated the efficiency and effectiveness of this man-machine interface sufficiently that non-CBI users are beginning to adopt them.

Generally the CBI market for hardware is small and must follow the trends set by other large users. The trend which CBI followed -- from maxi-computers to micro-computers -- could scarcely remain unnoticed. The computer industry's development of smaller devices has not stopped with micro-computers, however. 1981 was the year that several other manufacturers announced hand-held computers (HHCs), and several other manufacturers began upgrading their most powerful calculators to become hand-held computers. Although some of the machines available are unthinkable as devices for delivering CBI as we know it, others are really rather flexible and provide some instructional opportunities not previously available. Computationally, one HHC contains a 6502 microprocessor, the same as used by Apple, PET, and Atari. That HHC claims to be faster than an Apple and to use a more efficient language.

Because the first half-dozen HHCs offered are so powerful, it seems reasonable to ask, "Will CBI developers embrace HHCs as warmly as they embraced microcomputers?"

### Hand-Held Computers Available

To understand the advantages and disadvantages of using HHCs for CBI, it's helpful to first examine the capabilities and physical characteristics of several HHCs (Table 1 and Figure 1).

Panasonic/Quasar. The essentially identical Quasar and Panasonic models are notable for their exceptionally fine man-machine interface design and well-prompted software. For example, when first turned on, a menu of available options is presented, one item at a time, labeled with the key to access the option. If application or language ROMs have been attached anywhere in the system, they automatically appear in the menu. Selection of an option gives further sub-menus until the desired program or function has been reached. Hitting the CLEAR key moves the user up the menu one level. The speed with which menu items are changed is controllable by the user at 10 speeds. A HELP key explains the purpose of any key.

Because of the large array of peripherals, Panasonic/Quasar provides three sizes of "trays" which make the HHC package rigid. The largest tray holds six peripherals and is 12"x17". This is really larger than "hand-held" or "pocket sized", but many of the peripherals important for instruction would add only 4" to the length of the HHC mainframe.

Many of the most powerful features of this HHC are available only using the SNAP language (a compact, threaded language). This is normally written on an Apple computer and downloaded. Based on addressing limitations, the maximum RAM or ROM possible is 4 Megabytes. More practically, however, the current I/O bus expander has only six ports. Each RAM peripheral will add 16K and each ROM peripheral will add up to 64K (four interchangeable modules of 16K).

Sharp/Radio Shack. The essentially identical Radio Shack and Sharp HHCs are unique in that they utilize two microprocessors. Also in contrast to most of the other HHCs, they are not expandable via application modules or memory extenders. All preprogrammed material must be entered via the cassette.

For a HHC with a full QWERTY keyboard, this device is very lightweight and thin. Its display consists of character areas, not a continuous matrix; graphics and special characters cannot be made. Like the Panasonic/Quasar, this HHC will store a BASIC programming statement that is 80 characters long, even though the number of characters that can be viewed in the display is only about one-third that many. The version of BASIC is fairly complete for arithmetic calculators, but it has no powerful string-parsing commands. The device suffers from low RAM storage capacity and lack of interchangeable ROM capability.

(Author's note: After this manuscript was submitted, an improved HHC was announced by Radio Shack. Although no detailed specifications or actual models were available for examination at the publication deadline, information in press releases indicates many of the shortcomings of the original product have been remedied. In any case, the information above and in the attached table refers only to the first model issued.)

HP-41C(V). This HHC looks like a calculator on the outside, and like a computer on the inside. For example, it has a full alphabet, but not in QWERTY layout. It is programmed in Reverse Polish Notation (RPN) logic, not a language. It has keys for many functions, but these functions can also be spelled. Inside it can do moderate string manipulation and can even print user-designed characters on its printer. The peripherals available include three printers, a magnetic card reader/writer, a wand for reading bar codes, cassette recorder, internal clock, and interfaces for video and to RS-232C and IEEE-488 ports. Four ports let the user connect ROM, RAM, or peripherals. Since earlier calculators' magnetic cards are compatible, a large library of routines and a considerable user group is available. Unfortunately, the magnetic cards are comparatively fragile for field use; the bar code reader provides a reasonable input device, however.

TI-59. The TI-59 is much more clearly at the calculator end of the spectrum. It has limited alpha capability, no programming language, and a simple and small display. As a programmable calculator it is fast, efficient, and (with overlays) simple to use.

Casio, Newbury, and several other manufacturers produce what might be considered HHCs, but the four devices described above are representative of what is available.

## REASONS FOR USING HHCs

The four main reasons for considering HHCs as CBI delivery devices are:

- portability
- capability for piggybacking or embedding training
- size
- cost

### Portability

Portability of training is probably the best reason for using an HHC. Though micro-computers are often called personal computers, HHCs may be considered even more personal; they can be carried in one's pocket in most cases and in one's brief case in all cases. Replaceable or rechargeable batteries provide at least 100 hours of power. Thus, it is easy to imagine never being without a computer more than a few seconds away. This concept is so new, it's implications are still not clear. What is clear is that at least one of the manufacturers, Panasonic/Quasar, has realized the potential "addiction" that could result and has incorporated features which would tend to "addict" users. This device has a real-time clock, alarm, and message system which lets one set multiple alarms for up to 139 years ahead and attach a message to each. A user of this HHC should never again forget a meeting, birthday, or medication. The same device lets the user create and search files he has constructed: phone numbers, "tickler" files, etc.

If an HHC is always with the user, the training applications are endless. Even with HHCs that are not regularly carried with the user, there are at least two good applications of the portability capability of HHCs: field training and salvaged time. Often trainees must walk around or be a passenger in a vehicle to perform their duties. In such cases there may be a considerable advantage to a portable training system. Moving great distances is not necessary: a lab or shop technician being trained to perform a multi-step task in a large work area will find it much more convenient to carry an HHC (plus a flask or a machine part) from workstation to workstation rather than returning to a central, fixed CBI terminal after each step.

Salvaging wasted time can, in effect, add one to four hours of training time to a worker's day. By carrying an HHC loaded with training material, anyone who spends a lot of time waiting (e.g., a derrick operator, a salesperson selling to corporate executives, a security guard) can regain time which is otherwise either stressful (waiting to make a sales presentation) or boring (waiting for a crane to be needed).

### Piggybacking and Embedding Capabilities

Piggybacking training is typically a sound and cost-effective concept whether the computer system used is a maxi-, mini-, micro-, or hand-held. Because many HHCs have keys that can be reprogrammed for each application package, it is very feasible to place a keyboard overlay on an

HHC which does real estate or aviation calculations, exchange the read-only memory chip (ROM) so training rather than computation can be done, and commence training. Easy access to and familiarity with the computer tend to promote convenient training. The extra advantage of HHCs is the range of subject matter which they are expected to have will expand the range of training opportunities.

Embedded training is another capability which is feasible on other devices, but which may find more applications with HHCs. Embedded training is training which is an intrinsic part of a tool or machine (generally an "intelligent" device). When an operator of the intelligent device uses it, it may sense training is appropriate and branch the operator to a training routine. An oversimplified example: if the user of an office copier pressed the "COPY" button three times without inserting a new original, an intelligent copier could surmise the user is unfamiliar with the multiple copy controls and provide a short set of directions. Such a copier does not yet exist: intelligent copiers exist, but they would use an internal microprocessor for embedded training, not a separate HHC. However, a few of the existing machines in our factories and offices are intelligent. To upgrade these devices without retooling some manufacturers may find it cost-effective to add an HHC to an existing machine for monitoring and controlling purposes. Once this has been accomplished, embedding training in the monitoring and controlling programs is straightforward.

### Size

Even when a power cord is not a disadvantage, the tiny size of a HHC may offer advantages: no doubt that is why an HP-41C was chosen for use by space shuttle astronauts. Most situations requiring miniaturization are related to moving vehicles and have already been discussed under the "portability" heading; nevertheless, there is usually an advantage to consuming less space: a desk with a HHC on it is less crowded than a desk with a microcomputer on it.

### Cost

Smaller devices have, in general, meant smaller prices. This has been one of the reasons for the success of micro-computers. The HHCs now on the market list in the $170 to $250 range. The Panasonic/Quasar lists for $500 - $600. These prices are very low; however, as with micro-computers, peripherals can raise the price to nearly $2,500.

## REASONS FOR IGNORING HHCs

The portability of HHCs places severe constraints on the keyboard and display. No pocket-size HHC has a keyboard on which one can easily touch type, although it is barely possible on the Panasonic/Quasar. A standard-size keyboard would limit portability considerably. Some HHCs use an ABCDE-format keyboard with a 10-key numerical layout which may be easier for some types of users to operate. However, no matter which HHC is considered, it is clear that entering large amounts of information is slower and less desirable on an HHC.

The same constraints that make the keyboard small limit the size of the display. Information presentation in the traditional mode of a CBI tutorial is not practical. It would be practical to use other portable media (e.g., printed matter) to provide background information, rules for simulation and games, etc., and to perform the interaction on the HHC. Within the space constraints of the display, the dot matrix screen provides much more flexibility for generating graphics and user-designed characters. For reasonable graphics, however, a flat TV image is needed. Animation of the existing displays is limited by the speed of the microprocessor and the persistence of the LCD display.

## CONCLUSION

The tradeoff between small size (for portability) and large size (for better keyboards and displays) is not clearly resolvable via technological advance. I believe the severe display constraints will restrict the CBI role of HHC at least until voice input and output capabilities are added. I feel that initially HHCs will be used by the military and by industry in those cases where mobility of the training device is paramount.

### Acknowledgement

The author gratefully acknowledges the support of the Army Training Development Institute, Ft. Monroe, Virginia, for the support of research which contributed to this report. The views, opinions, and findings in this report are those of the author and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

TABLE 1. TABLE OF HHC FEATURES

| | Panasonic/Quasar | Sharp/Radio Shack | HP-41C (V) | TI-59 |
|---|---|---|---|---|
| Height | 8-15/16" | 6-7/8" | 5.7" | 6-3/8" |
| Width | 3-3/4" | 2-3/4" | 3.1" | 3-1/4" |
| Depth | 1-3/16" | 11/16" | 1.3" | 1-1/2" |
| Weight | 14 oz. | 6 oz. | | 8.5 oz. |
| Microprocessor | 6502 | 2 4-bit custom | | components |
| Physical keys | 65 | 55 | 35 | 45 |
| Keys and shifts | 140 | 72 | 130 | 92 |
| Display | LCD matrix (5x8) | LCD matrix (5x7) | LCD segment | LED 7 segment |
| No. characters displayed | 26 | 24 | 12 | 12 |
| No. different characters | 96 | 58 | 59 | 20 |
| Upper/lowercase | full, plus inverse | upper | upper plus 5 l.c. | printer only |
| Internal RAM | 2K or 4K | 1.9K | 41C: .4K 41CV: 2.3K | .1K variables plus 1K |
| Interchangeable ROM | 64K or 48K | Not available | 32K | 5K steps |
| Selective dot erase | yes | no | N/A | N/A |
| No. sounds | 4 octaves | 1 | 10 | 0 |
| Duration control | full | 2 lengths | | N/A |
| Price | 2K: $500 4K: $600 | $170 C: $250 | C: $250 CV: $325 | $200 |
| Print matrix | 7x9 | 5x7 | 5x7 | 5x7 |
| Peripherals | application ROM | cassette | application ROM | application ROM |
| | 16K RAM | | magnetic card reader | magnetic card reader |
| | Acoustic modem | | clock/calendar | |
| | Color TV adapter | | bar code wand | 12V dc adapter |
| | RS-232 and IEEE-488 interfaces | | 3 printers | |
| | Disk | | 4K RAM | |

FIGURE 1. THIS HHC USER IS RECEIVING A DRILL AND PRACTICE LESSON THAT WILL FAMILIARIZE HIM WITH A COMPETITOR'S NEWEST PRODUCT OFFERING

# Integrating Simulation Activities with Instruction: Guidelines for Developers

David R. Abernathy and Kathleen A. Terry

Bell System Center for Technical Education

## ABSTRACT

The computer has two major abilities which are important to the instructional designer: to present instruction, and to simulate an environment in which desired skills will be performed. This paper presents guidelines for determining when each of these functions can be effectively introduced into a CBT course and when simulations should be integrated with instruction. These guidelines define several distinct categories of CBT applications. Finally, six major design requirements are associated with the corresponding CBT categories to which they apply.

## INTRODUCTION

This paper presents a view of major design decisions that are made during a technical education course development project when considering the use of computer simulations in a computer-based training sequence.

For instructional designers, several forces make the application of computers for simulating skill environments (eg., jobs) an increasingly attractive strategy. The number--and management level--of people who interact daily with computers in their work is increasing. Moreover, the ability of computers to effectively manage media and the number of available delivery configurations (mainframe and stand-alone computers) continue to grow. At the same time, the use of the computer to provide simulations of skill environments within instruction brings powerful instructional tools, as well as great challenges, to the instructional developer. The authors will present a checklist of design guidelines that have evolved over a two-year period in developing courses for telecommunications software maintenance engineers, circuit designers, and other technical managers (for remote delivery via the Control Data PLATO system).

The concept of integrating instruction and simulation contains implications for both instructional and computer programming design. It also represents the most expensive application of computers for instruction. The discussion that follows suggests that a CBT project will result in a more effective training product when developers carefully plan the computer's role in 1) providing instruction, 2) providing simulations, and 3) integrating instruction with simulations.

## DEFINING TERMS

For the purposes of this paper, the terms "simulation" and "instruction" are seen as clearly distinguishable applications of the computer for training.

"Simulation" refers to a modeling of outward appearances and responses to user actions of a given target environment--computer system, in-basket, or other stimulus set--within given limits. "Instruction" refers to the presentation of help, in the form of information, practice exercises, or prompts, which is intended to occasion the exercise of a skill in the normal environment (for example, on the job).

## SIMULATIONS

Table 1 relates training objectives to use of computer simulations. Simulations are not efficient when, in the normal (work) environment, the skills are used in a variety of settings. For example, general technical skills training

| TRAINING OBJECTIVES: | USE OF COMPUTER SIMULATIONS: |
|---|---|
| Skills are NOT applied in a specific environment.<br>EXAMPLES:<br>General technical skills training. | Computer simulations are NOT used. |
| Skill involves a set of relatively independent actions; "correctness" can NOT be inferred from a specific action.<br>EXAMPLES:<br>Troubleshooting and problem-solving courses. | Computer simulations are SEPARATE from instruction. |
| Skill involves the application of a relatively WELL-defined procedure.<br>EXAMPLES:<br>Long case problems; introduction lessons. | Computer simulations are INTEGRATED with instruction. |

Table 1. Use of the Computer for Simulations

in subjects such as AC/DC circuits and mathematics may focus on basic concepts which are applied in diverse ways in the work environment. To simulate just one or two applications would improperly narrow the scope of the training and yet greatly increase the cost of the development project. And to attempt to simulate all the applications of the skill would be impractical or impossible.

LEVEL OF INTEGRATION

The second section in Table 1 describes training products where computer simulations are relatively separate activities from the instructional material. Here the skill may involve a number of judgments, where at various points several actions may be equally appropriate in solving a problem. The simulator provides feedback that closely matches the real-life environment, while the instruction explains, summarizes, and ties together the series of actions taken in the simulator. For example, one course simulates the input and output messages of a computer system, while extended explanations of several interactions, including trouble-shooting procedures, are available as separate activities. Thus, students can manipulate the simulator in much the same way they would interact with the real system, and at their option switch to an instructional sequence which provides extended feedback on working the assigned problems. The key to this category is that the simulated environment is required to support a variety of relatively independent actions.

The third category in Table 1 includes training of clearly-defined procedures. Computer simulations are integrated with instruction. For example, a segment of instruction on editing data in a computer database requires the student to complete specific fields of data in a simulation displayed on a computer terminal. Correctional feedback is presented after each input. Integrating simulations with instruction is effective for training of relatively long, deliberate procedures because of the importance of providing immediate feedback for performance. And introduction lessons can provide a simulation of the entire course. The student completes a brief sequence which demonstrates the visual formats that the student will encounter and explains the student actions required to proceed through the course. (The importance of providing such an introductory module is greater for self-paced and remote delivery training situations than for courses with on-site instructors.)

| Course Characteristics | Amount of Instruction to be Provided by the Computer is: |
|---|---|
| A large primary target population exists.<br>The training equipment is similar to the real-life equipment used by the student.<br>Modular design serves varying skill levels.<br>Modular design is used to teach different target populations and objectives. | High |
| Existing materials are effective.<br>Other media are effective.<br>Methods to be trained change frequently.<br>The budget is limited. | Low |

Table 2. Use of the Computer for Instruction

## AMOUNT OF COMPUTER-BASED INSTRUCTION

Table 2 presents a general profile of the major factors which determine the amount of instruction which can efficiently be presented by the computer (versus off-line instruction). These factors include both "internal," instructional design criteria, and "external" criteria such as the extent of funding. Table 2 is directed at one specific use of the computer: to present instruction. Because this function can often be performed in other ways at lower cost, the developer must attempt to predict the delivery pay-off for basing each segment of training on the computer, regardless of the computer's role in providing testing, media management, and simulations.

## DESIGN REQUIREMENTS

Table 3 provides a table of requirements to be met by designers of training applications in each of the six categories which result from applying both of the earlier decision tables. The types of training applications are:

1. Low amount of computer-based instruction; no simulations.
2. Low amount of computer-based instruction; simulations are separate from instruction.
3. Low amount of computer-based instruction; simulations are integrated with instruction.
4. High amount of computer-based instruction; no simulations.
5. High amount of computer-based instruction; simulations are separate from instruction.
6. High amount of computer-based instruction; simulations are integrated with instruction.

To use Table 3, the developer first decides which of the above categories best describes the training being developed. The developer then applies the guidelines which are bulleted under the chosen category in Table 3. The major guidelines are distributed unequally over the above training applications. In fact, the number and cost of CBT course design features steadily increases with the use of simulations and the integration of simulations with instruction. The major requirements are:

1. Maintain the basic conventions used elsewhere in the course or in the target setting (computer key conventions, and consistent graphics and terminology) and eliminate barriers to student access to information (such as unnecessary sequence and review restrictions).
2. Provide tools. Examples of tools are calculation aids, practice exercises, sample tests, extended feedback, animations, interactive dialogs, inquiry databases, glossaries, and games. The upper limit for this activity is the point where the student experiences information overload; where the marginal return on the additional development effort is no longer profitable.
3. Define the limits of the simulation; match the characteristics of the instructional and real-life equipment. The goal is to

| Amount of instruction provided by computer: | LOW | LOW | LOW | HIGH | HIGH | HIGH |
|---|---|---|---|---|---|---|
| Computer support of simulations is: (SEP=separate from/ INT=integrated with instruction) | NONE | SEP | INT | NONE | SEP | INT |
| 1. Maintain conventions; eliminate barriers to student access to information. | ● | ● | ● | ● | ● | ● |
| 2. Provide tools (to the limit of their usefulness to the student). | | | | ● | ● | ● |
| 3. Define the limits of the simulation; match characteristics of actual/simulator. | | ● | ● | | ● | ● |
| 4. Teach the student how to use the simulator. | | | | | ● | ● |
| 5. Provide access between simulator & lessons; prescribe study based on performance. | | | ● | | | ● |
| 6. Make the simulator "verbose" and responsive to skill level. | | | | | | ● |

Table 3. CBT Design Requirements Based on Use of Computer Simulations

present an informative, usually simplified model to the student at a reasonable cost. Once the limits are defined, they must be explained to the student. For example, in the one course, students are told that only certain input message formats are accepted by a computer terminal simulator. Examples of acceptable formats are provided to the students, and formats that are not recognized by the simulator in practice are rejected as gracefully as possible by the simulator program. Matching the characteristics of the real-life and training equipment involves certain trade-offs. For example, the developer does not want to confuse the student regarding the appropriate response required. An early version of an introduction lesson written by one of the authors contained a simulation of the course table of contents index display that was so realistic that several students called the course delivery hotline complaining that the index display would not allow them into their lessons. In fact they were looking at a captioned facsimile of the index.

4. Teach the student how to use the

simulator. The student is given expanations or simple drills on using the simulator.

5. Provide access between the simulator and instructional material; refer the student to an appropriate segment of instruction based on the student's performance. In one course, the student manipulates a graphics display which is subsequently scored. The student receives a study prescription which identifies the page numbers in the work binder containing information on the requirements the student did not meet during the simulation.
6. Make the simulator "verbose" and responsive to individual differences among students. Here the simulator has built-in intelligence which is able to provide instructional feedback to the student's response, as well as the response expected from the environment being simulated. Although this format seems attractive, the developer has to be concerned with several potential problems. If, for example, there is more than one correct way to solve a problem in the simulator, provisions have to be made to accept an alternate path. One also has to avoid contaminating the simulation model with messages that mask the simulation. Third, the student will be unnecessarily detained if there is too much help given, or if the smallest unaided step is too small. Finally, the instructional prompts must be withdrawn at some point for the student to interact at a mastery level. Clearly, the developer struggles with establishing an appropriate context for the interaction, a constraint that is still fairly expensive to implement in CBT today.

Table 3 suggests that CBT products differ distinctly in the number of design elements required to develop them. In fact, more than one format may be required within a given course development. As mentioned earlier, introduction lesson segments may be an appropriate place to use simulations that are integrated with instructional material. Therefore, the tables can profitably be applied to chunks of instruction within a course. The primary goal of the paper is to clarify the set of choices that are available and ought to be considered by the course designer/analyst.

# Computer-Based Training

# (CBT): Buying Criteria

W. J. Hudson

Control Data Canada, Ltd.

## ABSTRACT

The purpose of this article is to list some criteria that should be considered before purchasing any computer-based training product/service. Also, to point out some negative economic ramifications that result from not considering this or a similar list of criteria.

This article is based on the following assumptions First, most buyers have never purchased a CBT product/service (CBT = Computer-Based Training/Education). Second, there does not presently exist a set of criteria buyers can use to make this decision. Third, most buyers work for companies/institutions that do not fully understand all present and future training/education needs.

To accomplish our purpose we will cover: normal buying criteria; questions raised when considering a CBT purchase; specific CBT buying taboos; considerations before making a CBT purchase and, finally, the possible negative economic ramifications that result when you do not.

At the end there is a checklist of questions you may wish to use before making a CBT purchase.

## NORMAL BUYING CRITERIA

Before making a major buying expenditure, most companies and/or institutions, have a set of criteria on which they base their decisions. Some criteria looked at are: how will this expenditure effect short-and long-range profit, how does this product or service compare with the competition, what is the return on the invested monies, what else could we invest our money in that would bring a better return, how will we use this acquisition to solve our present and long-range needs? - just to mention a few. Note that the companies/institutions have made a list of buying criteria, that they have some knowledge of the product and/or service they are buying and that they understand how the product and/or service will satisfy present and future needs.

However, when a new product/service is introduced one cannot assume this. If buyers are not familiar with a product/service, and have never made a similar purchase, they do not have a list of criteria, they do not have a good fix on how their present and future needs can be impacted, and they do not even know what items they should measure to make a sound economic decision. This is particularly true when most companies and/or institutions decide they will invest in computer-based training/education.

Few people truly understand computer-based training/education. Therefore, when they consider a CBT purchase they are left with more questions than answers.

## QUESTIONS

What is Computer-Assisted Instruction? What is Computer-Managed Instruction? Should we purchase a mini, a micro and/or a CBT system? What are the factors we should consider before buying? How can the CBT products/services solve my present and future training needs? This list goes on and on.

This article would have to be much longer to answer all these questions. In an earlier article (April 1981), I discussed what CBT is and its benefits. Here I would like to give you a list of criteria to consider. However, before we cover the buying criteria, we should examine some pitfalls associated with the purchase of a CBT product/service.

BUYING TABOOS

Do not base your decision on hardware! Do not buy until you understand how a CBT product can impact your present and future training needs! Do not buy unless the cost of a CBT product/service can be offset by a CBT solution of an identified need and/or offers enough flexibility to solve a possible future need! Do not buy unless you understand how the software behind the hardware limits your ability to solve education/training problems! Do not buy unless you are convinced the CBT vendor is committed to computer-based training/education! Do not buy a product to solve education/training needs unless the vendor has the personnel to help solve these needs! Finally, do not buy until you have investigated the total capabilities of CBT product/service you are purchasing and how it will impact your present and future education/training needs.

These taboos will become apparent as we cover and explain what you should consider before making a CBT purchase.

BUYING CRITERIA

Before making a CBT purchase, there are five major considerations: your present and future training/education needs, the hardware capabilities, the depth and sophistication of software, the courseware library and the vendor's expertise/commitment to the field of education/training.

EDUCATION/TRAINING NEEDS/SOLUTIONS

You must understand what you are trying to accomplish with the purchase of a CBT product/service. If you do not understand your present and future training/education needs, you cannot possibly make a good buying decision. You may purchase a product that takes care of a present need only to find out that your next need cannot be solved by the hardware you've purchased. Or, perhaps your purchase, with compromises, may solve a portion of your future need. In any event, the product you buy has to match your needs now and those that may arise in the future.

If your company/institution is like most, you may understand many of your present needs but do not know all future needs. Therefore, you should consider buying the most flexible computer-based training/education product/service.

A flexible product/service offers you:

- o The ease of authoring.
- o An available courseware library.
- o A sophisticated software package.
- o A variety of hardware capabilities.
- o A support and service group that understands hardware, software, courseware and the education/training world.

Once you have determined your needs you must determine the appropriate training/education solution. Most needs are best solved by a variety of media and methods. The product/service you are buying should offer you flexibility in solving different needs in different ways. Do not consider only the hardware price when you purchase! Your purchase should allow you a solution via text, video, audio, computer-assisted instruction and/or computer-managed instruction. In addition, the CBT product should be capable of the accepted teaching modes of inquiry, simulation, gaming, tutorial, and drill and practice.

HARDWARE CAPABILITIES

While hardware is certainly not the only buying consideration, it is an important one.

To enhance your ability to solve training needs the terminal should be designed for easy man-machine interface. Remember, many people who you may want to train have never seen a terminal before. Therefore, it's important that they're trained on a terminal designed for non data processing people.

In addition, a terminal's capabilities must enhance the learning process. It should be capable of rapid response time, quality graphics, easy-to-use input devices and selective erase. Also, the terminal needs the capability of operating standard peripheral equipment -- such as audio, video and printing equipment. This will allow you to choose different media for different educational/training objectives. Incidentally, a terminal should be able to run these shared peripherals without the need for design of a special interface unit.

DEPTH & SOPHISTICATION OF SOFTWARE

The CBT product you are purchasing should have software tailored for educational purposes and not for data processing. If it does not, it may not give operational priority to interaction with teacher or student. In addition, it should not have a language that was designed for data processing. If it does, you will find that authoring will be hindered and that you may be forced to make many pedagogical compromises when designing a program to solve a training need. The software must have a language that is easy to use for the non data processing person.

Furthermore, the software should have a command structure that allows the programmer a variety of choices in important areas such as answer judging, animated graphic presentation and variety in feedbacks.

Next, the software should incorporate useful utility programs -- utility programs that allow you to collect data, generate reports and communicate with peripheral equipment.

If any one of these features is missing, you may be hampered in solving a training need with the hardware you have purchased.

Hopefully, you can see that you need to understand not only your training needs and the hardware capabilities, but also, the depth and sophistication of software.

The remaining two important factors that impact a CBT buying decision are courseware and vendor expertise.

## COURSEWARE

Available courseware and your ability to create courseware are the most important factors in buying a CBT product/service. If you are making a CBT purchase to solve an education/training need it follows that the real solution lies in quality courseware -- a quality course that is already available, a course that you can modify and/or a capability that allows you to author your own quality course tailored to your specific needs.

Most CBT products offer some hardware and software capabilities but few offer you a large variety of courseware that can help with your training needs.

Developing a course is quite labour-intensive and costly. In fact, the cost to develop courseware is far more than hardware cost considerations. More specifically, you may be better off purchasing an expensive piece of hardware that has a courseware library with it than purchasing an inexpensive piece of hardware that requires you to design your own courseware.

Other courseware considerations are: how big is the vendor's present library; will the vendor continue to expand the library, and how many of these courses match up present and possible future training needs?

## VENDOR'S EXPERTISE AND COMMITMENT

Since the process of solving education/training needs includes needs analysis, design, development and production, the vendor from whom you purchase your CBT product/service should have more than hardware expertise. In fact, a necessary blend of expertise consists of personnel with knowledge of the publishing process, instructional design, software, hardware, and the learning process. In other words, you should look for a vendor who has educators, trainers, publishers and data processing people involved in their operation.

Just as important is the vendor's commitment to education and training. Is the vendor truly committed to the field or just dabbling in it? If not committed, the vendor may have a long-range plan calling for the use of technologies that make the current courseware obsolete.

In sum, you need to buy from a vendor who is actively involved in and will continue to be involved in the improvement and development of hardware, software and courseware for educational/training purposes.

Only after you understand all the aforementioned criteria will you know the true cost of your computer-based training purchase.

## NEGATIVE ECONOMIC RAMIFICATIONS

There are severe negative economic ramifications to buying a CBT product or service without considering the criteria we present here.

Two examples may clarify this. First, let's suppose you buy an inexpensive CBT product because it has an accounting application that can solve a present need. After surveying the marketplace, you make this decision because it appears to be the least expensive way to go. However, several months after your purchase, your company/institution realizes it has another training need. The inexpensive piece of purchased hardware does not have an application to solve this need. Therefore, you decide to design and develop your own course. The cost in doing this is far greater than if you had purchased a more expensive CBT product that had applications that could solve both needs. To develop one good hour of individualized, multi-media, computer-based instruction can cost you anywhere from $15,000-$30,000 per hour. Obviously, a vendor with courseware will be of more value than one with few or no courses.

Another example involves a company

that needs training in emergency safety procedures.

After evaluating two products, the company decides to buy the least expensive piece of hardware. However, after the purchase it is discovered that to build a course on this hardware is quite difficult. The language is awkward and forces the author to make many pedagogical compromises. In addition, a video device cannot be hooked up to this terminal without a special interface unit being built.

Unfortunately, the more costly piece of hardware they chose not to buy had both these features. That is, the software behind the hardware was designed for ease of authoring in training/educational situations. In addition, the terminal has a built-in outlet for a video device.

Here again is an example of the problems that arise when you purchase an inexpensive piece of hardware without considering all buying criteria.

The following checklst of questions should be answered before you make a CBT purchase.

PRESENT NEEDS

- Do I know what I want to accomplish by purchasing CBT?
- Do I understand the products' capabilities?
- Is it the best way to do it?
- Is it cost effective?

FUTURE NEEDS

- Do I know my future needs and if the CBT product/service that I am buying will be capable of solving those needs?
- Are the product/service capabilities varied and flexible enough that my unidentified future needs may be taken care of?

AVAILABLE COURSEWARE

- Does the CBT product I am buying already have solutions to some training/educational needs?
- How big a library does the vendor have?
- Will the vendor continue to publish more courseware?
- How many present courses match my needs?
- How many future courses will match my needs?

AUTHORING EASE

- If I have to write a customized course to meet a training need, how easy is it to author?
- Is the authoring language designed for training and educational purposes?
- Can the vendor train my people to author on the hardware?
- How does the hardware hamper authoring?
- Can I use CAI and CMI?
- Are learning activities assigned?

PEOPLE EXPERTISE

- Do they understand the hardware/software and how it impacts the learning process?
- Do they have people to help me identify needs; to design, develop and produce materials that will solve my training needs?
- Do they have people who understand publishing and are actively engaged in the development of a large courseware library?
- Do they have educators who understand the learning process?

DEPTH AND SOPHISTICATION OF SOFTWARE

- Was the software designed for data processing or for educational/learning process reasons?
- Does it allow for operations to give top priority to the teacher/student interaction?
- Is the language easy to learn for a non data processing person?
- Does the command structure allow the programmer a variety of choices in important areas such as:
  - answer judging,
  - animated graphic presentation,
  - variety in feedback?
- Does the software incorporate useful utility programs for:
  - trainee data collection,
  - report generation,
  - communication with and operation of peripheral equipment?

HARDWARE CAPABILITIES

- Are the terminals you're purchasing designed for easy machine-man interface?
- Does the hardware configuration allow for:
  - rapid response,
  - quality graphics,
  - easy-to-use input devices,
  - touch panel operation,
  - selective erase?
- Can the hardware operate standard peripheral equipment such as audio, video and printing equipment? Can the hardware do this without special interface equipment?

## VENDORS COMMITMENT

- How long have they been involved in training/education?
- Do their short-and long-range plans make use of new technologies without current courseware becoming obsolete?
- Is the vendor actively involved in the improvement and/or development of hardware, software and courseware?
- What portion of the vendor's total corporate commitment is given to CBT?
- Do they have people for equipment maintenance, software support, courseware development and educational consulting?

# GRAPHICS SIMULATION IN MAINTENANCE TRAINING--

# TRAINING EFFECTIVENESS AT COST SAVINGS

Brian Dallman

Air Force Human Resources Laboratory
Lowry AFB, CO 80230

## Abstract

This paper presents an overview of research the United States Air Force is performing in the area of simulation for maintenance training, with an emphasis on the use of Graphics Simulation. Factors such as reliability, expense, and safety make the use of actual equipment for training undesirable. The use of simulators presents a viable, cost effective alternative, but the best use of maintenance training simulation requires that the correct match between the level of simulator fidelity and the instructional task be determined. An Air Force research program is examining this issue, and the use of computer driven interactive imagery or Graphics Simulation promises to be an important fidelity level. It may deliver the best training for the least cost.

## Introduction

Computer-based instructional simulations have been successfully employed for many years. These simulations are particularly appropriate for learning objectives involving the understanding of processes and/or problem solving. In recent years the Department of Defense has invested heavily in research designed to provide information on the cost effective use of simulation for equipment operation and maintenance (i.e., trouble-shooting) tasks. This paper will discuss this line of research, focusing upon one promising form: the use of interactive graphics simulations for maintenance training. The papers which follow will examine other facets of interactive graphics simulation in detail.

## Need for Maintenance Training Simulation

Presently most Air Force Training is heavily dependent upon the use of Actual Equipment Trainers (AETs) to provide students with the necessary training experiences when learning to operate or maintain the equipment. There are numerous problems in using AET as a training device. For example, if the equipment is complex, reliability is often a problem. Frequent unskilled operator use often causes equipment breakdowns. Likewise, AETs are often expensive to acquire and maintain, too few may be available for student use, and in some cases there may be a safety hazard for the trainees. However, one of the most important limitations of AETs is that they are not designed to provide the required training experiences. Practice in fault diagnosis and extraordinary operational conditions is often limited by an inability to modify the equipment, prohibitive cost, or other factors. It is for these reasons that simulation has been receiving considerable attention as an alternative training strategy to the use of AETs.

After reviewing 12 studies dealing with the training effectiveness and cost of maintenance training simulators, Orlansky and String (1981) conclude that simulators come up the winner when compared to AETs. The study indicates there is no difference in the job performance of students trained either way, and the acquisition and maintenance costs are less for simulators. The lack of a difference in the observed job performance is a little mysterious in that one would expect better performance out of the simulator trained group. However, this seems to be more of a problem with the comparative evaluation design: only the capabilities common to the simulator and the AET were evaluated. So, in some cases, the simulator was designed to be just like the AET, and in others the instructional features were turned off or not evaluated during the comparative evaluation. In any case, the simulation is a promising instructional technique for maintenance training applications.

## Levels of Simulation Fidelity

The level of realism of a simulation is a characteristic generally referred to as fidelity. In maintenance training simulation "task fidelity" means to what extent the behaviors of a person in the simulated task match the behaviors of a person doing the real task. It refers to planning activities, interpreting information, control manipulations, and other cognitive and psycho-motor skills. Another often used term is "equipment fidelity", the extent to which the simulator looks like and acts like the real equipment. It is important to note that the task fidelity of the simulation will be influenced by the equipment fidelity of the simulator used. Also, task fidelity requirements should establish equipment fidelity.

Equipment fidelity can be thought of as a continuum. The highest fidelity would be the AET in a training setting, and the lowest could be a photographic mock-up or a picture drawing. In between there may be a three-dimensional simulator, a two-dimensional simulator, and an interactive image simulator. To distinguish between them is not always easy since there are relative degrees of each, but the 3Ds are made to look and act like the AET and have some physical representation of the internal components, the 2Ds usually represent the front panel of the equipment and will generally use cheaper representations of components. The interactive image simulator will use graphic drawings, pictures, photographs, etc., to represent actions the operator or maintainer might perform. In general, it can be said that the higher the equipment fidelity the greater the cost--you need a more powerful computer and more expensive software for the higher fidelity simulators. It is often thought that the higher the equipment fidelity the greater the transfer of training from simulator to actual equipment, but there is a problem: it is not true.

It is easy to push a button; it is more difficult to know which button to push. High fidelity equipment simulators may be necessary when learning a motor skill, but they are often not required when the task is principally a mental one. Tasks such as trouble-shooting faults, and performing operator activities are often more mental than procedural. Thus, equipment fidelity and task fidelity are not directly related. High equipment fidelity does not guarantee high task fidelity and AETs do not provide the best training experience for some tasks.

Because fidelity is a critical issue in simulation design, a principal research thrust in the area of maintenance training simulation is focused on the question, "For a given task which level of simulation fidelity is cost effective?" It is this question when considered with other factors (number of students, amount of training time, resources) that will drive the design of training system which meets its desired performance outcomes. In many situations it may be better to have 20 interactive image simulators and an AET, than two AETs, or an AET and two 3D simulators.

## Air Force Research in Maintenance Training Simulation

Recognizing the potential for maintenance training simulation the U.S. Air Force established a program which would produce information to guide the development, acquisition, and implementation of maintenance training simulators. Established in 1975, Project 2361, Simulation for Maintenance Training, has been the responsibility of the Air Force Human Resources Laboratory. The approach has been to develop and evaluate maintenance simulator training systems in actual training settings. The information produced by this research program and by other civilian and military organizations will be incorporated into a set of model specifications and user handbooks for the application of simulators to maintenance training.

A central line of research in Project 2361 is the fidelity studies. This is where the answers to the aforementioned question are being found. So far studies comparing a 3D, a 2D, and a photographic mock-up of the same aircraft avionics test stand have been completed. A final study employing computer driven interactive imagery, called the Interactive Graphics Simulation for Intermediate Level Maintenance Trainer, will be complete by January 1984. This effort is particularly interesting in that the highest task fidelity for the least dollar may be achieved using the Graphics Simulator and AET in combination.

## Graphics Simulation

The Graphics Simulation project will develop an interactive imagery research and development test bed to examine a number of important research issues. Of course a principal concern will be to determine for what tasks this technique is appropriate. Also, a number of instructional features which have important cost or training effectiveness implications will be examined (Peiper & Richardson, 1981). Institutional concerns, reliability, maintainability, and acquisition costs will also be examined.

The design of the simulation attempts to maximize task fidelity by representing the entire task. Everything from planning steps to equipment operation will be incorporated within the simulation. The simulation will be designed to blend with the instructional environment to minimize implementation problems and maximize student performance. The simulator will consist of a color graphics and a video disc adjunct media device terminal connected to the computer-based instructional facility at Lowry Air Force Base. A video disc unit was selected because of its ability to emulate most other types of media (sound/slide, motion picture, video tape).

The complex nature of today's military equipment requires better trained maintenance personnel. The Department of Defense budgets $18 to $20 billion dollars each year on maintenance activities alone (Orlansky & String, 1981). A cadre of poorly trained maintenance technicians can be financially disastrous and reduce the ability of a nation to defend itself. For the military, the investment in Maintenance Training Simulation will produce important rewards. The civilian sector will receive benefits as well by applying the knowledge gained by this research.

## References

Orlansky, J., & String, J. Cost-effectiveness of maintenance simulators for military training. Proceedings of the third interservice/ industry training equipment conference and exhibition. Orlando, FL: 1981.

Pieper, W., & Richardson, J. Graphics simulator detailed research plan. Lowry Air Force Base, CO: Air Force Human Resources Laboratory, 1981.

# AN INTEGRATED DESIGN AND DEVELOPMENT SYSTEM FOR GRAPHICS SIMULATION

J. Jeffrey Richardson

Denver Research Institute/SSRE
University of Denver
Denver, CO 80208

## Abstract

This paper presents three enhancements to a basic, straightforward, fixed sequence simulation system. The principal application of the resultant system is in the training of maintenance and operations technicians. The principal object of simulation is the task to be performed. This includes both the planning knowledge and the equipment actions required in performance of the task. The first enhancement facilitates the development, refinement, modification and use of a hierarchical task analysis of the job; the second enables the system to adapt its presentation strategy on an individualized basis; the third provides a means of integrating the conceptual and procedural knowledge related to the performance of the job being simulated.

## Introduction

The basic contribution to the field of maintenance and operations simulation by the system described below is the recognition that the principal object of simulation is the task to be performed. This includes both the planning knowledge and the equipment actions required in performance of the task. Prior maintenance and operations simulations have focused solely on simulating the equipment upon which the task is performed. The design elaborated below is a novel and unique departure from previous approaches to maintenance and operations simulation.

A secondary contribution is the use of graphics and images as the medium of representation for the simulation. When CAI simulation is coupled with graphics displays, the resultant graphics simulations provide a high degree of both physical and psychological fidelity. Thus, graphics simulation is a powerful tool for equipment-oriented operation and maintenance training. The graphics capabilities embodied in high resolution color graphics terminals, video disc images, and/or computer modified video disc images provide for realistic equipment representation and manipulation, as well as representation of the processes and concepts important in task performance.

The graphics simulation system is based on the capability to confront the student with a fixed sequence of decisions and equipment actions. The correct decision/action at any given step in the simulation causes the system to advance to the next step. Decisions are input either as choices from a list of alternative answers (multiple choice format) or as natural language input (constructed response format). Equipment actions are input, as nearly as possible, by graphics simulation of actual equipment actions, utilizing video disc images of the equipment.

Each simulation step begins with the presentation of text and graphics which set the scene for the planning decision or equipment action pending. These scenes often summarize the status of the problem to this point. When an inappropriate decision alternative or equipment action is chosen by the student, a feedback message is presented, followed by a prompt designed to help the student make the correct choice. The student must then try again. The simulation is completed when the student correctly responds to the last simulation step.

The type of simulation just described is not a "free play" simulation. A free play simulation would permit the student to pursue any sequence of decisions and equipment actions irrespective of its correctness. Rather, a "fixed sequence" system is described. Such a system has greater instructional utility than a free play system because (through use of corrective feedback and instructive prompts) the optimal, suggested or recommended approach to a task is developed and reinforced. However, the fixed sequence may include several alternative solution paths for each simulation so that all correct approaches are supported.

The simulation scheme developed thus far places no constraint on how the various steps within a simulation are related; nor upon how steps in different simulations of the same job might be related. These constraints are typically provided by instructional designers through some form of task analysis and instructional planning.

Here is an example of such a design decision. Suppose a series of simulations of the same job is being designed. The level of cueing and prompting might be withdrawn in later simulations. This corresponds to design decisions regarding input response format (multiple choice or constructed response). The multiple choice input presents the student with a recognition memory task; the constructed response input, a recall memory task. The two differ in that the recognition task provides cueing to the student, while the recall task does not. A fixed sequence simulation scheme can accommodate options regarding input response format, but does not provide design guidance in this area.

Here is a second design example. Again, assume a series of simulations is being developed. Such a series, if well designed, should be coherent. That is, each simulation should present the same set of choices to students when they are faced with the same task or subtask. In other words, the job logic implicit in each simulation should be consistent throughout all simulations. The fixed sequence simulation system described so far can certainly accommodate such consistency; but again, the system offers the designer no assistance in constructing coherent simulations.

A significant portion of instructional materials development effort is expended making such design decisions. Two enhancements to the basic system thus far described provide design assistance to the simulation developer. One helps in the development and use of a task analysis of the job in providing coherent sequences of simulations; the other uses this coherence as a means of modeling student performance, which is in turn used to individualize and adapt simulation presentation to each student's profile of expertise.

## Instructional Development Assistance

Task Analysis Development. The development of a quality task analysis can be assisted by an on-line interactive exchange of information between each simulation sequence as it is being developed and a global hierarchy of decisions/actions. This requires that a global hierarchy of decisions/actions be represented on-line. It also requires that each simulation's sequence of steps be represented on-line.

The global hierarchy is a means of representing the alternatives available to the problem solver at each step of the way toward a solution. The correct alternative is determined by the sequence of steps which make up the simulation itself. The simulation provides the sequence of steps; the global hierarchy, the alternatives at each step.

Neither the global hierarchy nor the simulations provide a representation of the rules or conditions under which a given step must be chosen. Indeed, the simulations are developed to illustrate these rules--to make the students face alternatives and apply the rules. The simulation system is primarily a practice medium, not a medium of expository instruction. Students must come to the simulations with a rudimentary understanding of the decision rules of the job. Their understanding then grows through experience and the corrective feedback and prompting provided at each step of a simulation.

In the hierarchy, a task is represented as a node; major subtasks as subnodes. The process is repeated. Each node in the hierarchy is successively refined into subnodes until a level of decomposition is reached where (for the given task under analysis and the given competencies of the learners) no further decomposition is needed. For maintenance simulation, the terminal nodes in the hierarchy are equipment manipulations. The upper part of the tree represents the range of planning alternatives that go into selecting various equipment manipulations at the lower part of the tree.

The exchange of information between a simulation under development and the global hierarchy would proceed as follows. The designer is at a given step in a simulation. The global hierarchy automatically provides a list of steps which have immediately followed that step in prior simulations. If one of the steps on the list provided is appropriate for the simulation under development, it is selected as the next step and accordingly incorporated into the on-line representation of the developing simulation. If the list does not provide an appropriate alternative, the designer adds a step to the simulation under development. This causes, as a side effect, the new step to be incorporated into the on-line representation of the global hierarchy. If adding to the global hierarchy causes any contradiction or other difficulty for the already developed simulations, the system interrupts the development process and informs the developer of the trouble. The developer then makes appropriate modifications to the global hierarchy and prior simulations affected by the change.

Task Analysis Use. The primary benefit of the above mechanism for task analysis development results directly: the series of simulations automatically has coherence. There is consistency in the size of the simulation steps within a given simulation and consistency in the steps used across a series of simulations.

Another design area facilitated by this enhancement is a means of giving students meaningful multiple choice alternatives to choose from at decision steps in the simulation. For any node in the hierarchy, the set of immediate subordinate nodes serves as the list of multiple choice items for that node. Design efficiency also results because a given multiple choice menu, applicable to possibly dozens of simulation steps, need never be explicitly entered by the developer. This is done by software at run time. If a menu change must be made during the development process, it is made only once to the master copy (e.g., in the global hierarchy), and not dozens of times. This capability facilitates hierarchy refinement by reducing the effort associated with making changes. Coherence between simulations is again promoted because simmilar steps in different simulations present the same set of menu items.

The use of the global hierarchy has profound instructional implications: it is an organized list of instructional objectives. It serves to familiarize the student with a set of decisions/actions which constitute the job being learned. When familiarity with the global hierarchy and its application is complete, as evidenced by consistently correct responses on instances of each global node, the student will have mastered the job.

While the design aid presented here helps in the development and use of a quality task analysis, the designer must still ensure that a balanced set of simulations is chosen.

## Student Modeling

A second major limitation of the basic simulation system is that it provides no structured means of individualization. As used here, individualization means two things. First, it means accommodating changes in student expertise as learning occurs by appropriately varying such things as the level of cueing provided. Second, it means accommodating differences in rate of learning which exist between individuals. It is most desirable to individualize at the level of the smallest unit of analysis. In the present case, this corresponds to the level of the global node (i.e., each discrete decision/action step within the overall task hierarchy for the job).

Individualization has two requirements. First, a source of variation in presentation format is needed--one cannot individualize if there is nothing to change in the instruction provided. Second, given such variation, a means of choosing the appropriate presentation level for a given global node for a given student is needed. Each of these considerations is discussed in turn.

Variation in Presentation Format. The basic system provides the option of presenting a step in either recognition or recall format. Supplying the variation required for individualization is a very simple matter: either a menu is displayed or it is suppressed. In both cases, the correct answer is the same. However, in one case, the student need only recognize the correct answer; whereas, in the other, it must be recalled.

Further variation in presentation is possible. After a student has mastered equipment actions in the recall mode, the equipment action can be presented in summary form (e.g., the action is performed for the student and the results are summarized). Once a student knows how to set up a voltmeter, it is not necessary to practice this skill each time it is encountered. For planning steps, additional variation is supported by permitting the student to skip planning steps interceding between the current location in the simulation and the next equipment action. This "look ahead" strategy is a recognition that as students become familiar with a procedure, planning steps are performed automatically and need not be explicitly required.

Choice of Presentation Format. Given the option of presenting any given step in any of the above modalities, a means of choosing the presentation level for each step for each student is needed. No such means would be possible if simulations were not coherent internally and externally. Student modeling is accomplished through use of the global hierarchy. Experience (encountering the same step in different situations) and expertise (correctly responding to steps at the level at which they are presented) is recorded on a copy of the global hierarchy. When a step is to be presented, the associated node in the hierarchy is referenced, the student's expertise with this node is read, and an appropriate presentation level is chosen.

The sequence of presentation levels for equipment actions is recognition, recall, and summary. The sequence of presentation levels for planning steps is recognition, recall and look ahead.

The student model equates presentation level with expertise. "Recognition" is the least expertise for a node; "summary" or "look ahead," the greatest. A criterion is established for shifting the expertise level up or down. For example, five consecutive correct responses to steps corresponding to a global node might be the criterion for shifting the level of that node from recognition to recall. Similarly, perhaps one incorrect response would cause a node at the recall level to be reset to the recognition level. When all of the nodes have been shifted to their highest level due to consistently correct responses of the student, the job may be considered mastered.

Relationship Between Enhancements. Individualization accomplished through student modeling requires both a global hierarchy and variation in presentation format. The task analysis design aid presented above provides both of these key elements.

The goal of any instructional design effort is instructional effectiveness and efficiency. Tailoring the presentation of instruction to the characteristics of an individual helps promote both of these goals. Instruction in areas in which the student has known expertise may be avoided--promoting efficiency. The resultant concentration of instructional effort on areas known to need instruction results in effectiveness given that a quality task analysis of the job has been performed.

## Conceptual Knowledge Hierarchy

This enhancement is presented last because it has no implications for the previous enhancements discussed. The knowledge hierarchy is a means of connecting two aspects of instruction which are often left unreconciled: theory and practice; or, conceptual knowledge and procedural knowledge. All of the explanations one might give of what a job is, how equipment operates, what resources are available to the problem solver, etc., which are not attached to a particular problem but are offered as general information, can be thought of as theoretical explanation. Theoretical material is usually the type of material contained in student workbooks and the type of material the instructor puts up on the blackboard during lecture.

On the other hand, there are, for the steps in any given simulation, explanations about each step which could be offered, often the feedback messages or prompts. A slightly more general explanation might be offered as a help sequence.

At some level of detail, say at the level of detail of the help sequence, the information presented could be used either as problem specific explanation or general theoretical information. The terminal nodes in the knowledge hierarchy contain these dual purpose explanations. The hierarchical structure above these terminal nodes is used to organize them into a corpus of explanations structured around topics in the curriculum.

Each knowledge hierarchy node is accessed in one of two ways: from the root of the hierarchy or from a step in a simulation which contains a pointer to that node. If the node is accessed as a help from a simulation, the student could, if desired, explore other related nodes by moving through the hierarchy. If the node is accessed from the hierarchy itself, it may be possible, in principle, to run the associated simulation steps as examples.

As simulation step help sequences are designed, one first notes whether there is an appropriate point in the hierarchy to reference. If so, a pointer to this reference is incorporated into the step. If not, a help sequence is developed for the step and then integrated into the hierarchy. Similarly, conceptual information is archived into the hierarchy directly. Hence, lecture materials supply potential help sequences; help sequences round out the lecture material. The hierarchy represents both sets of explanations as one. They are useful as lecture support if entered from the outline perspective, and as simulation support if entered via the pointers installed in each simulation step.

## Summary

The principal object of the simulation is the task to be performed by the maintenance technician. This includes the planning knowledge and equipment actions required in performance of the task. When CAI simulation is coupled with graphics displays, the resultant graphics simulations provide a high degree of both physical and psychological fidelity. Thus, graphics simulation is a powerful tool for equipment-oriented operation and maintenance training. The graphics capabilities embodied in high resolution color graphics terminals, video disc images, and/or computer modified video disc images; provide for realistic equipment representation and manipulation, as well as representation of the processes and concepts important in task performance.

The system under development can be viewed as a basic fixed sequence simulation augmented by two levels of enhancement. The first level provides for the development and use of quality task analyses as follows. One comprehensive, integrated set of simulations is to be created. As a simulation is developed, each decision/action point is cross referenced to all similar steps made in other simulations through the mechanism of a global task hierarchy. This interactivity ensures a consistency and coherence of simulation steps, both within and between simulations.

A student model is formed by the use of the global hierarchy to record student experience and expertise across simulations. This model drives a set of selection strategies which promote instructional efficiency and effectiveness by focusing training on those decisions/actions known to need attention.

A knowledge hierarchy helps integrate the conceptual and procedural knowledge related to the job simulated. Terminal nodes in this hierarchy accessed as help frames from simulation steps provide explanation in the context of the problem being solved. These nodes are organized and subsumed as a hierarchy which, when accessed from the top, provides a conceptual explanation of the job, organized around topics.

It is one thing to have an authoring system which permits building sequences of simulation steps. It is quite another thing for such a system to provide design assistance. The enhancements described provide this type of guidance, making the development of quality simulations less critically dependent on the expertise of trained instructional designers.

## References

Pieper, W.J. & J.J. Richardson. Graphics simulator detailed research plan. Lowry Air Force Base, CO: Air Force Human Resources Laboratory, 1981.

Pieper, W.J. & J.J. Richardson. Graphics simulator interim report. Lowry Air Force Base, CO: Air Force Human Resources Laboratory, 1981.

# GRAPHICS SIMULATION SYSTEM RUN TIME OPERATION

William Pieper

Canyon Research Group

Denver, CO 80208

## Abstract

This paper discusses the run time characteristics of a Graphics Simulation system designed to present the opportunity to practice both the planning and equipment manipulation tasks for technician jobs. Since the system is data driven, each of the data structures is discussed in terms of its content and use. The display characteristics and devices for both the cognitive/planning information and equipment manipulation activities are discussed, together with simulation episode scenarios. Finally, the paper discusses the operation and administration of the simulations, including enhancements which adapt simulation performance to the state of each student's competence.

## Run Time Overview

The simulation run time operation is controlled by an executive program which manages those data structures supporting student practice of relevant job behaviors. The principal "data structure" is the Task Evaluation Network (TEN). The TEN defines the planning activities and equipment manipulations the student must accomplish for each separate problem. A related structure is the Global Task Hierarchy (GTH) which defines all the tasks in a single job (i.e., the collected TENs).

Each simulation encountered by the student consists of a series of specific events (termed episodes) presented on one of two display devices. The primary device is a color graphics terminal of medium resolution, with keyboard and touch screen inputs. The adjunct medium is a television monitor with videodisc, disc controller, graphics overlay system, and touch screen input. The primary graphics terminal is used to present scene setting material for the episode, activity menus, feedback, and prompts. The adjunct medium is used for simulating equipment manipulation.

An executive program controls the flow of the simulation including the extent of student cueing through a menu of alternative actions for each episode in the simulation. All of the student's accomplishments are recorded in a Student Model, and the level of cueing provided the student is based on this model. If a student makes an inappropriate choice during an episode, the system provides both feedback concerning the "wrong" choice and a prompt to cue the student to the correct alternative.

It is important to keep in mind that this simulation system does not engage in the presentation of declarative CAI, it is designed to provide the opportunity to practice job skills in a learning environment. The function of instructing the students remains the responsibility of the instructors. The graphics simulation system described here is designed to support the instructor's activities, not supersede them.

## Run Time Data Structures

Task Evaluation Network. Each TEN defines and completes simulation. The data structure is a linked list of direct access files. Each item in the list consists of an episode identifier (name), an episode type, one or more forward links (children), one backward link (parent), and references to other constituent elements of the episode (e.g., feedback messages, prompts, scene material, and directives to the adjunct medium).

Global Task Hierarchy. This structure is a hierarchical tree of all tasks included in the specific simulations (TENs). The function of this structure is to record the generic tasks performed by the incumbent in a form similar to that of a job/task description. Its principal use is in presenting a menu of alternative task choices for each episode. This structure is a tree with nodes consisting of 20 character alpha strings (node name), a pointer to a named predecessor node, and one or more pointers to named successor nodes.

Knowledge Hierarchy. The knowledge hierarchy items contain information and drawings used by instructors during lecture. The data structure is a list of direct access text and graphics files. The items in the files are used in scenes, prompts, and as help when the student is having trouble. The information provides visualizations of the process going on in the equipment and descriptions of the state of the equipment.

The knowledge hierarchy files are linked together so that they may be examined as an integrated body of information. Once this hierarchy is entered, the items may be examined by following the links established at the time the hierarchy was created. Moving upward in the hierarchy brings the user to more global unifying concepts and moving downward in the structure provides finer levels of detail on specific topics.

Student Model. This data structure is simply a copy of the Global Task Hierarchy items in list form. The information contained in each item is as follows:

a. the current menu level of this task,
b. number of successful attempts of this task at the current menu level,
c. total times the task was encountered across all simulations to date.

This data structure is the only one which is modified during run time. Its primary purpose is to provide information to the simulation executive for use in establishing the menu level to be used by the student during each episode. The simulations based on the above data structure are presented on the graphics terminal and adjunct medium devices.

## Graphics Terminal

The following information and interaction is presented on the graphics terminal.

Scenes and Menus. The items for presentation in each simulation episode are the same for all episodes. At the start of each episode, a scene is presented on the graphics terminal with the desired equipment items presented on the adjunct medium. During the initial scene, the conditions of the simulation problem are presented along with a representation of the current system configuration. At the completion of the scene, a menu of possible next actions is presented, and the student is simply asked what should be done next.

The student enters his or her choice by typing in the desired key words as presented in the menu. Typing key words was chosen as the mode of response in order to begin to shape the student's vocabulary. If the choice is correct, the simulation proceeds to the scene for the next episode. This process is continued for each of the planning episodes through the complete simulation. As long as the student does not make an error, the scene for each successive episode will be shown, followed by the menu of tasks which could be performed next.

Feedback and Prompts. If the student incorrectly chooses a task as the next one to be performed, he receives a short message explaining why that choice was incorrect. Feedback messages are specific to each incorrect alternative. Once a feedback message has been given, a prompt is provided to help the student arrive at the correct choice on the next attempt. The prompt looks very much like a scene; however, it is more specifically answer-oriented.

Help and Knowledge Review. The student has another choice if difficulties are being encountered. Help can be requested, whenever it is available on the menu screen. Pressing the HELP key will cause material used during lecture on this topic to appear on the screen. This material is part of the Knowledge Hierarchy. The student can study the material and when finished return to the menu screen from which he entered the help sequence. The knowledge hierarchy material can be studied separately, without going to it from a simulation by means of a help sequence.

## Adjunct Medium

The adjunct medium provides all the equipment manipulation simulation. The images of the equipment are stored on a videodisc which is controlled by a microcomputer. In addition to the disc and controller, there is a graphics system which can modify the equipment visuals at presentation time. The image modifications are stored as files on the microcomputer's disc drives.

All presentation of the equipment is done on a standard resolution color video monitor fitted with a touch screen. The touch screen is used by the student to indicate which of the equipment items, controls, and indicators are of interest during the equipment manipulation activities. If the student wishes to see a particular equipment item of the test station, he or she touches the area in which that item is located on the overview of the station. The view showing the quarter of the test station synonymous with the area touched appears. The student next touches the exact panel desired and the view of that panel appears. Each of the controls or indicators to be examined or manipulated is touched in turn, after which the view of that single control appears. A control can be manipulated or "read" only when it is shown singly on the screen.

During presentation, a typical sequence of images would be as follows:

a. an overall view of the test station,
b. a view of the device front panel (e.g., signal generator),
c. a view of one of the controls (e.g., frequency selector),
d. a view of the control set in the selected position,
e. a view of the next selected control and setting,
f. a view of the next selected equipment item and controls and settings.

Even though this simulation technique is capable of supporting free-play simulation, it does not. Students are not permitted to set irrelevant controls, view irrelevant indicators, or work on irrelevant equipment items. This approach was chosen because of a decision to reinforce only desired behavior patterns.

### Simulation Administration

The executive program is essentially a set of procedure calls. The functional organization of the program is depicted in Figure 1. The program reads several data structures but writes in only two. The TEN is the principal controlling structure, providing information regarding where in the GTH the menu should be obtained; what scene's, feedback, and prompts are to be used; what equipment simulations are to be used; and which knowledge hierarchy information items are relevant. The TEN is written into by the executive to record which of the tasks have been completed and therefore where the student is in the simulation. Information is also written into the Student Model to keep a running count of which menu level is to be used with each of the episodes.

The sequence of activities is as follows:

a. The executive reads the next node of the TEN and loads the following information:
   Scene displays
   Feedback messages
   Prompt pointers
   Equipment manipulation pointers
   Knowledge Hierarchy pointers
b. The menu is read from the GTH and loaded for presentation after the scene pages.
c. The Student Model is read for this node along with the current menu level indicator.
d. Presentation of the episode is started with the scene information.
e. Control is passed to the Knowledge Hierarchy driver as necessary.
f. Control is passed to the equipment manipulation program in the adjunct medium as necessary.
g. Menu level information is revised based on student performance.
h. Student Model information is updated by revising the count of successful attempts at the current menu level.
i. The TEN node completion index is updated.
j. Return to a. with the next node information.

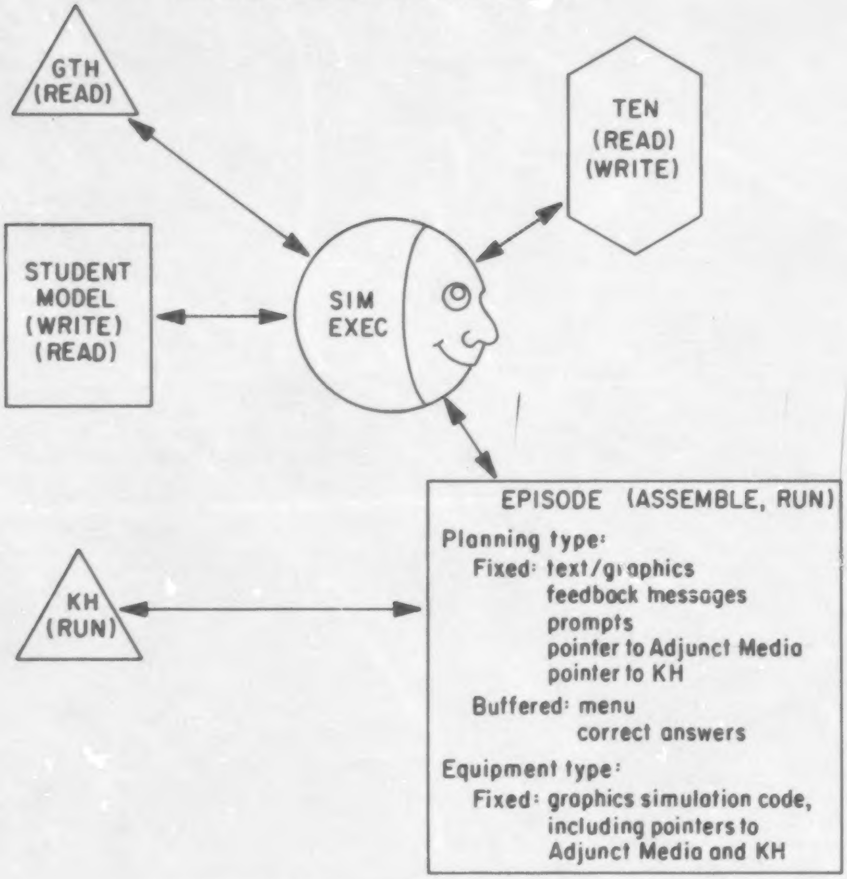
Normal Administration. The normal administration for the simulation is with a full menu available for the choices appropriate to each episode. This early menu availability provides response cueing and makes the task of deciding what to do next mainly a recognition task.

Likewise, the normal mode of presentation for the equipment manipulation activities requires the student to manipulate each of the controls and indicators for each piece of relevant equipment. While this would always be required on the job, it definitely lengthens the training time for each simulation. This normal mode does not take into account the learning that occurs. The student is required to manipulate equipment items on which he or she is facile as well as items on which he or she needs practice.

Menu Fade Out. The first step in modifying the decision task from one of recognition to one of recall is to fade out the menu. After the student has made a minimum number of successful tries at selecting the desired next task, the menu for the choices is no longer presented unless the student asks for help or makes an error. The executive program maintains a criterion level for fading out the menu and compares it to the menu level index for that episode in the Student Model.

If the menu is suppressed and the student asks for the menu, there is no penalty; the menu will be suppressed the next time the student arrives at this menu node. However, if the menu is suppressed and the student makes an error in choice of next task, there is a penalty. In this case, the menu level index is set back one level and the number of successful attempts at this level is set to zero. In effect, the student must requalify for having the menu suppressed.

Planning Look Ahead. The second level of menu modification is that of permitting the student to skip those planning tasks which have been successfully performed in the menu recall mode. In this case, the student is permitted to select, as the next task to be performed, (a) any planning task which is at either the recognition or recall level, or (b) any equipment



GRAPHICS SIMULATOR STRUCTURE--RUN--TIME

NOTE: Structures contain their name, and in parentheses, procedures which operate on them at run time.

Arrows indicate flow of information and control.

GTH
(READ)
TEN
(READ)
(WRITE)
STUDENT
MODEL
(WRITE)
(READ)
SIM
EXEC
EPISODE (ASSEMBLE, RUN)
Planning type:
Fixed: text/graphics
feedback messages
prompts
pointer to Adjunct Media
pointer to KH
Buffered: menu
correct answers
Equipment type:
Fixed: graphics simulation code, including pointers to Adjunct Media and KH
KH
(RUN)

FIGURE 1

task which follows one or more planning tasks all of which are at the look ahead level. In look ahead, the student may not bypass those planning tasks which are not at the look ahead level.

Equipment Summary. In order to speed up the simulations and provide more practice on equipment items not learned, an equipment summary capability is used. Once a student demonstrates that he or she is able to set up, connect, and read the results of a piece of test equipment without error, manipulation of that piece of equipment is no longer required. The simulation simply provides a summary of the set-up employed. The student must still identify which piece of equipment should be used and what signal is to be measured; however, manipulation of the equipment is no longer required. In addition, the student must still interpret the result to determine whether it is in or out of tolerance.

## Summary

The run time operation of the Graphics Simulation System utilizes several data structures to provide task menus, feedback messages, prompt scenes, help information, and equipment manipulation sequences to aid job task practice in a learning environment. The simulations are presented on two computer driven color graphics devices with the level of task selection cueing and equipment manipulation activities adjusted in accordance with student accomplishments.

# AN EDITOR FOR THE AUTOMATED GENERATION OF GRAPHICS SIMULATIONS

Anne Montgomery

McDonnell Douglas Automation Company
P.O. Box 30204
Lowry AFB, CO 80230

## Abstract

This paper describes software in the form of a dialog editor, for the automated generation of interactive graphics simulations. The Simulation Generation Editor allows a nonprogramming subject matter expert to design and produce simulations of equipment and procedures for instructional use. The specific target application was the Advanced Instructional System (AIS) located at Lowry Air Force Base, Colorado.

## Introduction

The high cost of actual equipment and 3-D simulators has caused increased interest in training and performance testing via computer terminals. While this type of simulation cannot completely replace the more realistic methods, it can provide the student with extensive practice before spending time on the actual equipment or a 3-D simulator, thus reducing the time spent on expensive resources.

The AIS Simulation Generation Editor was created to aid the author in the development of instruction involving dynamic simulation of equipment or processes. It was developed as an adjunct to the AIS CAI Authoring Editor which supports a variety of tutorial and drill-and-practice tasks but is inefficient for the development of interactive graphics simulations. However, many of the design concepts employed in the CAI Authoring Editor were expanded and incorporated into the Simulation Generation Editor.

## Design Goals

The overall design goal of the Simulation Generation Editor was to provide an easy-to-use authoring system which would allow nonprogramming subject matter experts to produce interactive, dynamic simulations. With few exceptions, CAI programming languages and development methods, employed to date, have required extensive and very detailed effort. Computer programming skills have been required as well as relatively sophisticated instructional design skills. These problems can be alleviated and CAI production costs can be reduced by providing software tools which eliminate the need for computer programming and which structure the authoring task. Not only do these tools directly reduce the time required for materials development but the imposed authoring structure reduces design skill requirements.

To achieve ease-of-use, we chose a prompting, dialog approach that provides the author with menus consisting of all actions that are appropriate at a given point in the simulation development. The rationale for this prompting approach was based on prior Air Force experience showing that sophisticated authoring tools were of no value if authors were reluctant to learn or incapable of learning how to use them. Therefore, emphasis was placed on a conversational format in which difficult concepts were implemented via a highly structured dialog that guides the author in a step by step manner through the process, thereby making the author more comfortable with the system.

## Approach

The approach taken to support the production of interactive graphic simulations centered around the development of a Simulation Generation Editor and a Presentation Program. The Simulation Generation Editor allows the user to define and store a simulation and the Presentation Program allows an interactive simulation to be "played back" for student use. Both the Editor and Presentation Program are designed to operate on most standard ASCII terminals.

## Simulation Structure

As structured by the Simulation Generation Editor, an interactive graphic simulation is sequenced into discrete tasks called events. While an event is a completely arbitrary designation, it should be thought of as a series of steps in a simulation that deal with a group of interrelated components. For instance, in a simulation of aircraft cockpit controls, an event could be defined to simulate a dial and its various readings or to simulate keyboard input and the resultant display changes.

The steps of an event are defined by a set of basic simulation functions called actions. Actions are performed on any of three types of objects--a graphic, a block of text, or a numeric. All actions can occur conditionally, as the result of an author-defined expression, or unconditionally, the default condition. The author can define as many as 50 events in a single simulation and each event can contain up to 125 actions and 75 defined expressions, conditions, or algorithms. A maximum of 200 objects can be defined for use across the entire simulation.

Throughout the Editor, a philosophy was adopted of providing extensive prompting and "fail safe" mechanisms. Each author entry is prompted to the extent possible. Commands that result in the deletion of materials require a second "Yes, I want to do it," response on the part of the author, and it is always possible to back out of an error situation without the Editor taking any action. Further, the Simulation Generation Editor continuously plays out the simulation actions as they are defined. For example, all terminal displays always appear as they would to the student. When the author defines an action, it is automatically executed following completion of the definition. This continual execution during development provides a validity check for the author to insure that the simulation is performing as intended.

## Simulation Objects

Three basic object types are used by the Simulation Generation Editor: graphic, textual, and numeric. Each object is identified by a one to ten character name assigned by the author. Graphic objects are created using the AID Graphics Editor. Textual objects are created by the author using the text editing/word processing facilities of the Simulation Editor. Numeric objects are author-defined variables, whose values are derived from the evaluation of author-defined expressions or from internal data kept by the simulation program.

## Action Types

When defining a simulation, the author may use any of eight types of actions:

1. DISPLAY/REDISPLAY
2. ERASE
3. STUDENT INTERACTION
4. GENERATE RANDOM NUMBERS
5. PAUSE
6. COMPUTE
7. REPEAT
8. BRANCH

## Action Description

The basic action used to create terminal screen displays is DISPLAY/REDISPLAY. This action allows the author to display, position, and size an object on the screen (Figure 1 shows a graphic object). For example, to give the effect of a moving dial the author would display a graphic of the panel and display the dial positioned at its initial reading. To move the dial, the author would then specify that the dial be redisplayed at an increment from its current position (see Figure 5).

Once an object is displayed, it can be removed from the screen using the ERASE action. The ERASE action allows the author to erase an object or a portion of the screen. There are four subtypes to this action (see Figure 4). The first, ERASE an OBJECT, erases an object that has been displayed on the screen by redrawing it in black. This type of erase is used to erase an object that is in close proximity to another object. The second, ERASE the DOMAIN of an OBJECT, allows the author to quickly remove an object from the display by erasing the minimum rectangular area enclosing the object. This action is much faster because it does not redraw the object. Both actions require the author to give the object name. If the object is not currently displayed on the screen, no action takes place. The third ERASE action allows the author to designate a portion of the screen to be erased. The final ERASE action erases the entire display.

The STUDENT INTERACTION action provides the author with a means for allowing a student to interact with the simulation. There are two subtypes of this action--TOUCH and KEYBOARD.

The KEYBOARD subtype is used to accept student responses from the terminal keyboard. The author is given the choice of providing his or her own prompting for input or using the default prompts provided. If the author chooses to provide the input prompt, the position on the screen where input is to occur must be noted. Next the author specifies a variable into which the keyboard input is stored.

The TOUCH subtype requires the student to interact with the simulation using a light pen or touch panel. The author must define touch areas on the screen (see Figure 2). Both correct and incorrect areas should be identified. When all touch areas have been entered, the author tells the Editor how to judge the student input. Two different situations are provided. The simplest case is when only one specific area on the screen is to be touched (e.g., simulation of a multiposition dial). A more complex simulation can arise when a series of areas must be touched either in a specific sequence or in any sequence (e.g., setting switches on a panel or pushing buttons in a specific order). Feedback can be associated with each touch area or touch sequence to provide students with substantive information about their responses.

Upon completing the STUDENT INTERACTION action, the author performs the action as if he or she were a student running the simulation.

The GENERATE RANDOM NUMBERS action is used by the author to introduce randomness into the simulation. The author is allowed to specify a range within which the random number is to be generated and an author-defined variable in which it is to be stored.

The PAUSE action allows the author to specify time intervals between actions occurring on the display.

The COMPUTE action allows the author to define expressions. Up to 75 author-defined expressions can be entered per event. The author enters the name of the variable (i.e., the numeric object) that will be the result of the computation. The equation to be evaluated is then entered. It is parsed and validated by the Editor. If there are no errors, the equation is evaluated based on the current state of the simulation. (Recall that the simulation is continuously being executed or played out as it is developed.) The value of the equation is then displayed, permitting the author to verify that all steps in the simulation are correct. This is illustrated in Figure 3.

The REPEAT action allows the author to repeat a sequence of actions any number of times during the simulation (see Figure 6).

The BRANCH action allows the author to branch from any action to any other action within the simulation.

## Simulation Scenario Script

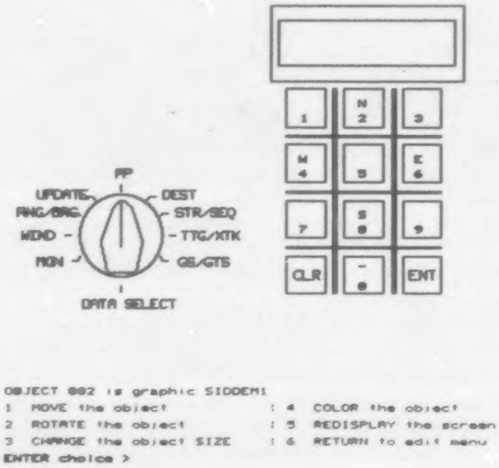
Because the definition of a simulation is a step-by-step process using a limited number of actions, the author should plan out the simulation scenario before using the Simulation Generation Editor. The use of a script, detailing what is to occur during the simulation can substantially reduce development time. Script writing entails a detailed, step-by-step analysis of the tasks that make up the simulation and a translation of these tasks into actions the Simulation Generation Editor can perform.

## Simulation Generation

The author's approach to creating a simulation is to define the objects (graphic, textual, and numeric) involved in the simulation, the object attributes, the actions by which the objects are to be manipulated, and the conditions under which the actions are to take place. Once an object is defined, the initial and current attributes of that object are maintained automatically. This information is then available to both the author and the Editor program, allowing the author to easily define interrelated processes without bothering with "bookkeeping" or programming details. For instance, because the program knows the domain of an object, the author can use the ERASE action to "erase the object." The author is not required to input or even know the location of the object. To further aid the author during the development, the simulation can be "played back" at any time during the simulation definition.

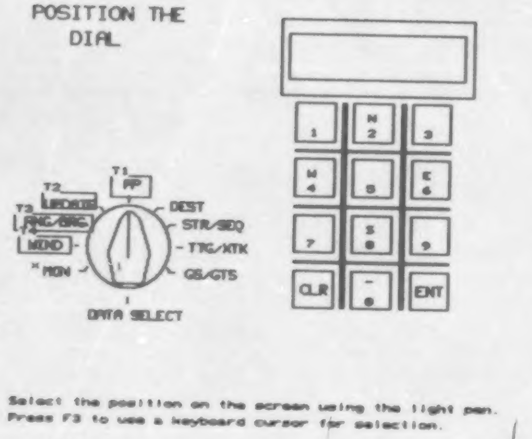
## Summary

The dialog editor approach to the generation of interactive graphics simulations appears to be quite promising. Subject matter experts with no programming experience can construct complex graphic simulations in relatively short time periods.



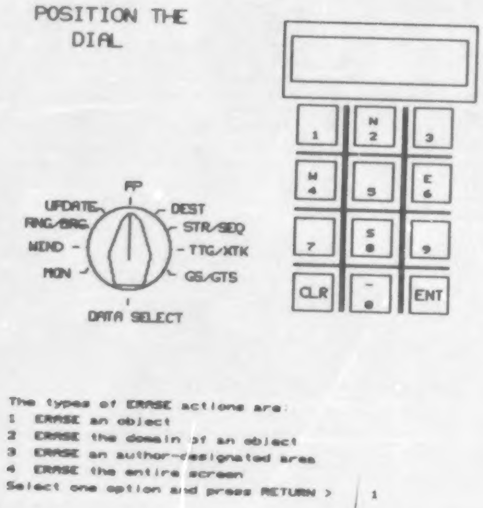
```
OBJECT 002 is graphic SIDDEM1
1  MOVE the object              : 4  COLOR the object
2  ROTATE the object            : 5  REDISPLAY the screen
3  CHANGE the object SIZE       : 6  RETURN to edit menu
ENTER choice >
```

Figure 1. Display an Object.


POSITION THE DIAL
T1
PP
T2
UPDATE
T3
RNG/BRG
WIND
MON
DEST
STR/SEQ
TTG/XTK
GS/GTS
DATA SELECT
1
N 2
3
W 4
5
E 6
7
S 8
9
CLR
-
ENT

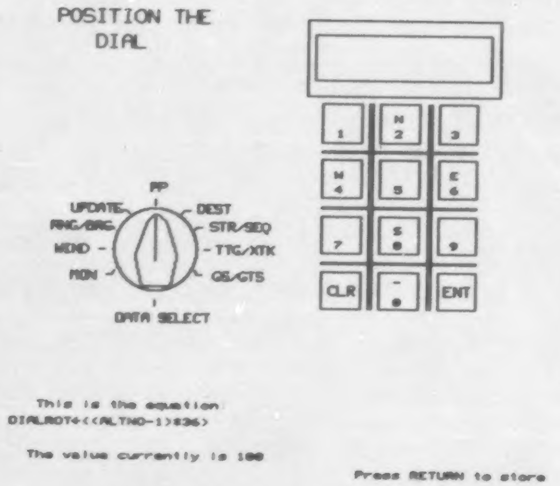
```
Select the position on the screen using the light pen.
Press F3 to use a keyboard cursor for selection.
```

Figure 2. Touch Student Interaction.


POSITION THE DIAL
PP
UPDATE
RNG/BRG
WIND
MON
DEST
STR/SEQ
TTG/XTK
GS/GTS
DATA SELECT
1
N 2
3
W 4
5
E 6
7
S 8
9
CLR
-
ENT

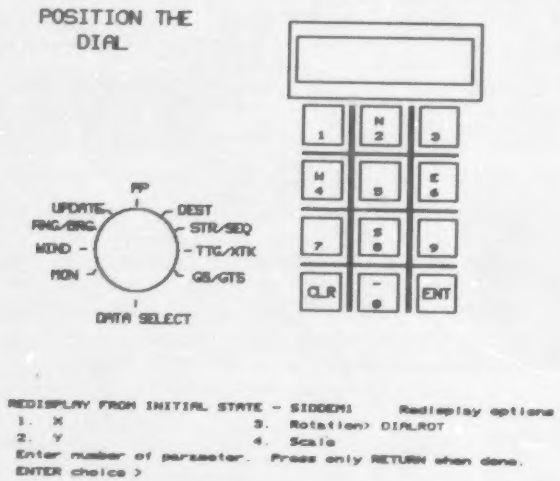
```
The types of ERASE actions are:
1   ERASE an object
2   ERASE the domain of an object
3   ERASE an author-designated area
4   ERASE the entire screen
Select one option and press RETURN >     1
```

Figure 4. Erase Action.


POSITION THE DIAL
PP
UPDATE
RNG/BRG
WIND
MON
DEST
STR/SEQ
TTG/XTK
GS/GTS
DATA SELECT
1
N 2
3
W 4
5
E 6
7
S 8
9
CLR
-
ENT

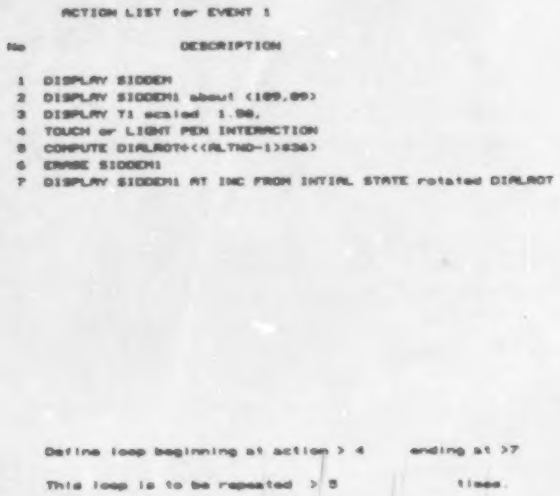
```
   This is the equation:
DIALROT+((ALTNO-1)*36)

  The value currently is 100

                                   Press RETURN to store
```

Figure 3. Compute Action.


POSITION THE DIAL
PP
UPDATE
RNG/BRG
WIND
MON
DEST
STR/SEQ
TTG/XTK
GS/GTS
DATA SELECT
1
N 2
3
W 4
5
E 6
7
S 8
9
CLR
-
ENT

```
REDISPLAY FROM INITIAL STATE - SIDDEM1      Redisplay options
 1.  X                           3.  Rotation> DIALROT
 2.  Y                           4.  Scale
 Enter number of parameter.  Press only RETURN when done.
 ENTER choice >
```

Figure 5. Redisplay Action.



```
      ACTION LIST for EVENT 1

No                  DESCRIPTION

 1  DISPLAY SIDDEM
 2  DISPLAY SIDDEM1 about (109,89)
 3  DISPLAY T1 scaled  1.50,
 4  TOUCH or LIGHT PEN INTERACTION
 5  COMPUTE DIALROT←((ALTNO-1)*36)
 6  ERASE SIDDEM1
 7  DISPLAY SIDDEM1 AT INC FROM INTIAL STATE rotated DIALROT




     Define loop beginning at action > 4     ending at >7

     This loop is to be repeated  > 5              times.
```

Figure 6.  Repeat Action.

# GRAPHICS SIMULATION HARDWARE--A TWIN DISPLAY SYSTEM UTILIZING A VIDEO DISC AND A MEDIUM-RESOLUTION COLOR GRAPHICS TERMINAL

Dale Steffen

Denver Research Institute/Electronics
University of Denver
Denver, CO 80208

## Abstract

This paper presents the design rationale and resulting system configuration for a computer-based graphics simulator. The simulator is presently being developed to provide a research vehicle allowing the investigation and identification of the effectiveness of computer-based graphics simulation techniques in the training of maintenance and operations technicians in the U.S. Air Force. The system configuration being developed is not intended to represent a final recommended operational device but rather will consist of a more sophisticated configuration also capable of collecting data that can be used to specify an operational, low-cost, stand-alone interactive graphics simulator.

The major components of the device consist of a computer, a color-graphics terminal, and adjunct media. The computer to be used is a large Air Force system known as the Advanced Instructional System (AIS). The adjunct-media device consists of a videodisc player and a microcomputer-controlled graphics generator which can superimpose graphics displays over the videodisc imagery. Touch panels on both the color-graphics terminal and adjunct-media display serve as interactive inputs for a student.

## Introduction

For several years the Air Force has been involved in the investigation of and experimentation with simulations for teaching equipment maintenance tasks (1).

The present state-of-the-art in computer and associated peripheral technologies strongly suggests that interactive graphics simulation may present a cost effective adjunct training device in a learning environment. In response to a perceived concept of the Air Force Human Resources Laboratory, such a simulator is presently under development to demonstrate its feasibility for supporting maintenance training of Air Force personnel. The objective of this project is to conduct research which will provide the Air Force with the necessary information to complete a more inclusive investigation being performed to identify both cost effective and training effective guidelines for the design and procurement of training devices which can be utilized for maintenance training.

In an effort to provide information for satisfying the more inclusive goals of the Air Force, the objectives of this project go beyond a demonstration of the feasibility for graphics simulators by, first, generating a research vehicle for investigating instructional strategies within the context of a low physical fidelity/high psychological fidelity device, and then secondly, using the device to conduct research which will supply the data necessary to determine the potential cost/training effectiveness of the various features inherent in such a training device. Thus, the activities of this project entail additional efforts other than the development of the necessary hardware and software which will result in such a research vehicle. Those efforts are presented in other papers (2, 3, 4, 5). The emphasis of this report is to present an overview of the actual graphics simulator being developed.

The system configuration must accommodate research issues associated with the interaction between a computer, a computer terminal, an adjunct media device, and alternate input devices via an interface controller. The baseline being followed in the development is to insure that the system will support all research issues to be addressed in the project. Thus, the actual configuration is not intended to represent a final recommended simulator system but rather will consist of a more sophisticated configuration capable of (1) representing a variety of feasible configurations, and (2) collecting data that will be necessary to evaluate and recommend an optimal configuration for an actual graphics simulator system. An example of the first point just stated is the decision to use a videodisc player to supply the video for the adjunct-media device. The result will be an ability to display: (1) visual images in still format in a manner similar to a slide projector or microfiche viewer; (2) visual images in a linear motion sequence in a manner similar to a filmstrip projector or video cassette player; and (3) any combination of the still and motion sequences as well as utilize the programmed branching capabilities available in videodisc players. An example of the more sophisticated capability to collect research data is the use of a large-scale computer system which already exists as an Air Force Human Resources Laboratory facility. This computer-based system, which utilizes Cyber 73-16 with associated software, is the Advanced Instructional System (AIS) and is capable of supporting a variety of instructional research and development efforts. As with the example of the disc player, the features of this system can be selected and utilized in a way that will represent a variety of computer capabilities to establish the requirements of actual computers that would be used in a stand-alone graphics simulator system.

With this baseline, performance requirements were established from the project research plan (6) to allow identification of the required functional characteristics of the simulator system configuration. These functional characteristics were then matched with the characteristics of several configurations resulting from various combinations of color graphic terminals, videodisc players, and alternate input devices (e.g., light/sonic pens, joysticks, touch panels) to identify a final configuration. The remainder of this paper is devoted to a discussion of the functional characteristics and configuration of the final graphics simulator.

## Functional Characteristics

The functional characteristics of the system are primarily based on the performance requirements of the two presentation devices, a color graphics terminal and the videodisc player. A keyboard or alternate input device(s) must provide a student with the ability to interact with the display of either presentation device in a simple and timely fashion. A discussion of the functional roles for each of these devices is presented to provide insight into the performance requirements and resulting functional characteristics. Discussions of the remaining components are based on the support of these roles.

A. Graphics Terminal Functional Role. The primary role of the color graphics terminal is to provide the task elements of a simulation. Two procedural components to be represented both explicitly and implicitly in simulation are (a) the job logic, and (b) the equipment tasks. In addition to the provision of these elements, the terminal will also provide declarative knowledge, theory, and functional information on an as-needed basis.

The fulfillment of this role will require a variation in the display format and the data displayed. For example, during a simulation, the display may be divided into several parts. One part could display a textual description of the system status, while a second part could display a functional block diagram of some portion of the system. A third part could contain a menu of actions to be taken. In the course of the student interaction via the keyboard or alternate input device, only some of these parts of the display may be changing. Also, more detailed presentations of an equipment component may be displayed for student interaction. These examples illustrate that textual characters should be resident in terminal read-only-memory (ROM) for rapid access and display, while high-graphics resolution should be available for line drawings. Requirements such as the presentation of block diagrams also dictate that graphics and textual characters be interspersed or capable of being mixed in the display. Finally, to provide time-sequenced simulation of equipment components in real-time, it is important that the access and display times of the complex graphics/textual displays to be presented on the terminal display be minimal. This suggests a local graphics buffer capability in the terminal to store graphic elements previously sent from the computer.

A summary of the performance requirements for the color graphics terminal to fulfill the simulation role just discussed is as follows:

1. The color terminal shall display different types of images on different parts of the screen, namely,
    a. One part of the terminal screen must display standard character sets derived from an ROM resident in the terminal.
    b. One part of the terminal screen must display graphic figures in an independent manner.
    c. One part of the terminal screen must display both graphics figures and standard character sets in varying sizes.
    d. An addressable cursor must exist to provide selective erase, write and sense for allowing noncoincident display changes on the various portions of the screen.
2. The color graphics terminal shall exhibit high-resolution graphics to provide quality representations of equipment.

3. The color graphics terminal shall have a minimum capability of displaying six colors plus black and white, a number determined in the research plan. Also, selectable background colors are desirable.
4. The requirements of the system which relate to dynamic graphics representation dictate that response time of graphic frame presentation be minimized. The implication of this requirement indicates a local graphics buffer from which graphic frames can be drawn and displayed upon demand under master program control residing in the AIS computer.

B. Adjunct Media Device Functional Role. The primary role of the adjunct media display is to represent comprehensively the actual equipment. Photographs of the actual equipment to be simulated will be displayed. These photographs will be structured to reflect and parallel the nomenclature of the equipment, from greatest to smallest detail. The photographs will also represent the various states of front panels within the simulations to be developed. No attempt will be made to represent these panels in every possible configuration.

The "rest state" for the Adjunct Media is a display of an overall view of the actual equipment. Protocols are being developed for student manipulation of this display. Through the use of an alternate input device (other than keyboard), the student can cause this display to show any area of the actual equipment at the necessary level of detail (e.g., a single rotary switch or a meter face). Interaction with the components of the actual equipment is simulated by the student accessing the picture of that component and then manipulating it through the alternate input device.

The above description presents a minimum level of capability for the Adjunct Media device. However, the following description illustrates a problem to be overcome with additional capabilities. In order for a piece of equipment with two or three position switches and four indicator lights, to be completely represented through pictures of all of its states, a total of 96 different pictures would be required. If on the other hand, the indicator light condition and switch settings could be modified on the TV screen through computer-generated graphics superimposed on the displayed image, only one picture would be required. In addition, some pieces of equipment contain "windows" displaying alphanumeric information. In order to simulate the possible states of this equipment on the Adjunct Media device, it is necessary to generate and superimpose characters of the appropriate size, shape, and color. Therefore, the capability for inserting vector graphic objects and ROM resident text characters or modified ROM resident text characters into a videodisc-generated display will be incorporated. An enhanced adjunct medium with computer-generated graphics under program control also provides cueing and other interactive techniques.

A summary of the performance requirements for the adjunct media device to fulfill the simulation role just discussed are as follows:

1. The adjunct media device shall be able to provide a comprehensive representation of the actual equipment. Thus, visual images derived from photographs of the actual equipment being simulated shall be presented on the display of this device.
2. Since the above requirement would result in a tremendous number of photographs to depict all possible system states (various knob positions, indicator lights, etc.) an additional capability of the adjunct media device shall be computer-generated graphics superimposed on the displayed image to simulate the positions of knobs or state of indicator lights and other equipment displays.

C. Alternate Input Device(s). With the description of the above functional roles as a context, it is now possible to discuss the considerations given to the selection of the alternate input device.

Touch screens were selected for use as the alternate input device since these devices represent the simplest type of operation by the student; that is, the use of a finger to identify a location on the screen. Also, if the touch panel selected provides the proper resolution, a stylus could be used by a student to simulate the equivalent operation of a light pen or sonic wand, thus providing a maximum flexibility to study the effects of "gesturing-action" input devices. If touch screens are affixed to both the graphics terminal and the color monitor used to display images from the adjunct media device, the student is able to interact directly with visually presented information. Such interaction does away with identifying a position on a display screen via a keyboard or other input devices that do not provide simple correspondence between the required input information and the displayed information.

Resolution along the X and Y axes will be adjustable into different ranges for different simulation applications. If interaction with fairly large segments of the display is desired as in a menu selection, a coarse adjustment can be used. Conversely, fairly small segments of the screen can be interacted with using finer adjustments. Thus, the coarser adjustment could be used on the graphic display terminal to allow the student to answer questions or to choose menu items by pointing at the screen without the requirement for high accuracy. The finer resolution could be used to allow the student to interact with the graphic images generated by the videodisc signals or graphics generator at the color monitor.

The performance requirements for the touch panels (as the alternate input devices) to operate in conjunction with the previously stated performance requirements of the color graphics terminal and adjunct media device are:

1. The alternate input devices shall provide adequate control resolution to allow necessary manipulation of graphically represented equipment on either the terminal or the display of the adjunct media device.
2. The alternate input devices shall be operable by "gesturing actions" from the student to minimize operational requirements for students.

D. Interface Controller. Including the interface controller and AIS computer to complete the graphics simulator results in a full configuration as depicted in Figure 1.

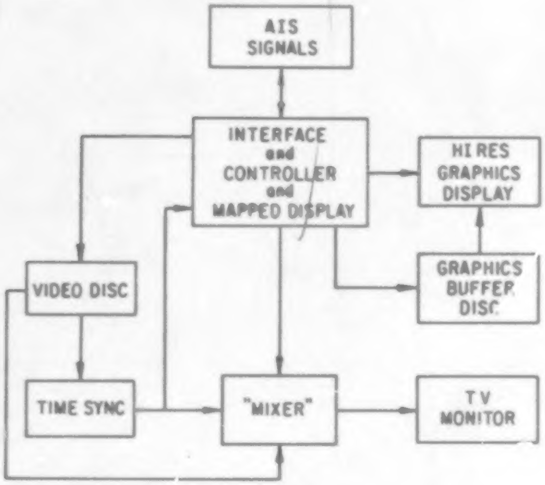


FIGURE 1. FULL CONFIGURATION

In considering the control of all the components to be used in the graphic simulator, certain basic operational or design parameters were followed. Each of these devices must either be controlled by the AIS computer or generate input data to the AIS. The major components to be interfaced or controlled are: (1) the color graphics terminal display; (2) the color graphics terminal keyboard; (3) the alternate input device for the color graphics terminal; (4) the videodisc player; (5) the adjunct media graphics generator; and (6) the alternate input device for the adjunct media color monitor. The components of the graphics simulator system have, except for the adjunct media graphics generator, RS-232C, standard ASCII protocol interfaces. The adjunct media graphics generator must be controlled by some type of microcomputer system, so it too can be effectively interfaced to the AIS computer via an RS232C standard ASCII protocol interface in the microcomputer. Thus, it was decided that all interfacing would be accomplished through the use of only one RS-232C interface port in the AIS computer and to create an interface controller in the graphics simulator system which would route communications to/from the AIS and each component of the simulator. Figure 2 depicts this configuration. Notice the configuration illustrates that the controller is a microcomputer and, in fact, it is the same microcomputer used to control the adjunct media graphics generator. As such, this configuration requires software development to ensure that the microcomputer will perform as required. However, such a software development effort will be significantly less than the software effort imposed on the AIS to control the adjunct media device. This configuration requires a minimum of effort in AIS software development to effect the necessary communications protocol.

The functional characteristics which result from the performance requirements and configuration as presented are given in Table 1. The list of equipment used for this configuration is presented in Table 2.

## Summary

A computer-based graphics simulator containing: (1) a color graphics terminal; (2) a videodisc-based adjunct media device exhibiting graphics overlay capability; (3) touch panel input capability for the displays of both devices; (4) a large computer-based instructional system (AIS); and (5) associated control and interfacing is being developed. The purpose of such development is to create a research vehicle to identify cost effective and training effective guidelines for the design and procurement of training devices which can be utilized for maintenance training.

The specific objectives of the research project which will utilize this graphics simulator are: (1) to demonstrate the feasibility of using interactive graphics simulation as an adjunct in a learning center environment consisting of a combination of actual equipment and three-dimension (high physical fidelity) simulators, (2) to investigate the training effectiveness of graphic simulation, and (3) to use data resulting from the research to develop a functional specification for actual low-cost stand-alone interactive graphics simulator.

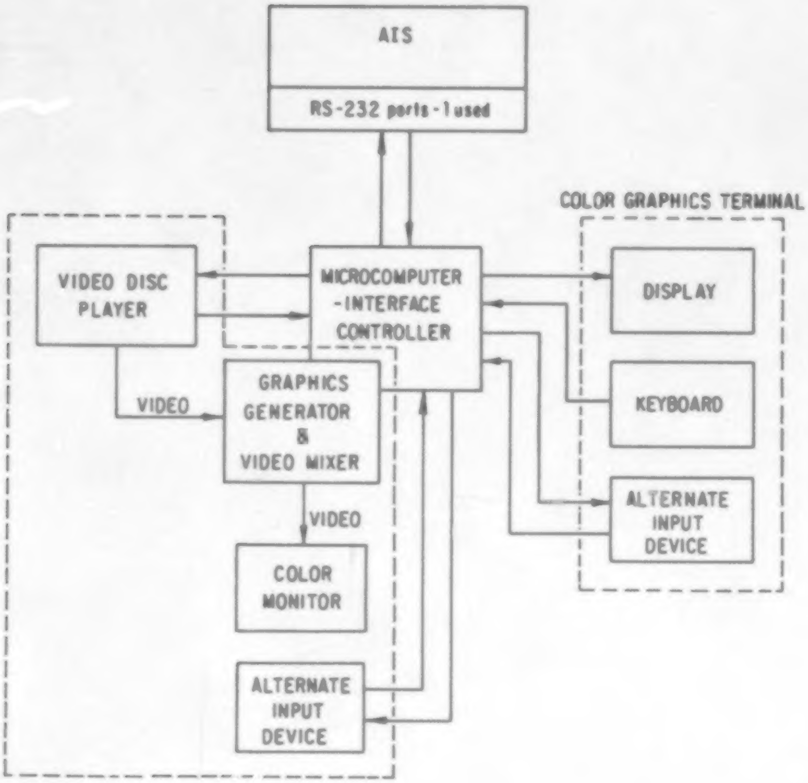

AIS
RS-232 ports-1 used
COLOR GRAPHICS TERMINAL
VIDEO DISC PLAYER
MICROCOMPUTER -INTERFACE CONTROLLER
DISPLAY
KEYBOARD
VIDEO
GRAPHICS GENERATOR & VIDEO MIXER
VIDEO
COLOR MONITOR
ALTERNATE INPUT DEVICE
ALTERNATE INPUT DEVICE

FIGURE 2. SINGLE INTERFACE TO AIS AND INTERFACE CONTROLLER FOR INTERFACING SIMULATOR COMPONENTS.

---

TABLE 1

FUNCTION CHARACTERISTICS OF GRAPHICS SIMULATOR

Graphics Display

1. Color display with a minimum of six color plus black and white.
2. Screen size small enough so that scan lines are not distractors.
3. Selectable background color at least a gray illuminated background.
4. Approximately 512 by 512 resolution.
5. Selective erase, write, and sense--addressable cursor.
6. No mode shift for text vs. graphics display (i.e., mixable)
7. One ROM'd character set so standard characters are not vector drawn.
8. Touch screen (minimum 32 x 32 resolution).
9. Graphics buffer disc.

Adjunct Media Devices

1. Color TV monitor display.
2. Size to represent 16" wide object near life size (i.e., 15 or 19 inch).
3. Videodisc with addressable frames through an RS232 communication port.
4. Disc control codes should require only ASCII characters.
5. Disc should be capable of motion display and still framing.
6. Speed of motion sequence should be variable.
7. Touch screen (minimum 32 x 32 resolution)
8. Screen mapping.
9. ROM'd characters placeable anywhere on the screen (still frame only).
10. Vector drawing--objects placed anywhere on the screen (still frame only).
11. Characters and objects displayable in color.

**Interface (Controller)**

1. Act as switch for signals received from AIS to send to Graphics Display, Graphics Buffer Disc, Videodisc, or Video "Mixer."
2. Transmit signals from "Terminal" to AIS.
3. Accept input for transmission from Keyboard or Touch Screen.
4. Store CGG for display through "video mixer."

---

### TABLE 2

### EQUIPMENT LIST FOR GRAPHICS SIMULATOR

**Graphics Terminal**
Chromatics CG1599 graphics terminal
32K RAM
Expanded Card Cage
Disc Controller/DOS
Floppy disk drive--dual sided
Hardware Vector Generator
Complex boundary fill software

**Touch Screens**
TSD TF-15H for Graphics Terminal
TSD TF-15H for Adjunct Medium

**Videodisc Player**
DiscoVision DVA 7820 II w/ RS-232
Videotek VM-15 color monitor

**Video Mixer**
Symtek PGS
Symtek Videodisc Controller

**Interface Controller**
Apple II+ microcomputer w/ 48K RAM
Floppy Disk Drive #1
Floppy Disk Drive #2
Language System with Pascal

---

## References

1. Miller, Gary G., "A Survey of the Potential Application of Generalized Simulation to Air Force Technical Training," AFHRL/TT, Lowry Air Force Base, (1976).
2. Dallman, Brian, "Graphics Simulation in Maintenance Training--Training Effectiveness at Cost Savings," 1982 ADCIS Conference Proceedings, (1982).
3. Richardson, J. Jeffrey, "An Integrated Design and Development System for Graphics Simulation," 1982 ADCIS Conference Proceedings, (1982).
4. Pieper, William, "Graphics Simulation Run Time Operation," 1982 ADCIS Conference Proceedings, (1982).
5. Montgomery, Ann, "A Graphics Simulation Display Editor," 1982 ADCIS Conference Proceedings, (1982).
6. Pieper, W.J. and Richardson, J.J., "Graphics Simulator Detailed Research Plan," Lowry Air Force Base, CO: Air Force Human Resources Laboratory, 1981.

# Relationship of Performance on a Computer-Based Context-Free Fault Diagnosis Simulation to First-Semester Performance in the Aviation Mechanics Curriculum

Karen S. Gaddis
William B. Johnson
Stanley R. Trollip

Aviation Research Laboratory
University of Illinois at Urbana-Champaign

## ABSTRACT

A study was conducted to assess the value of a computer-based fault diagnosis simulation as a predictor of student performance in an aviation maintenance curriculum. The simulation was adapted for use as a test and administered to twenty-seven first-semester students enrolled in the curriculum. Correlation and multiple regression procedures were used to evaluate the predictive utility of the simulation relative to that of a mechanical aptitude test currently used. Results indicated that measures of performance on the simulation provided predictive information comparable to that of the mechanical aptitude test.

## INTRODUCTION

The Aviation Mechanics Curriculum (AMC), established by the Institute of Aviation at the University of Illinois, is a three-year terminal curriculum consisting of coursework that is both technical and liberal in content. Hands-on experience in performing maintenance on aircraft and engines is emphasized in preparing students for mechanic certification by the Federal Aviation Administration. Essential to the role of the aircraft mechanic, and thus critical for success in the AMC, is the ability to detect and diagnose system failures.

Results of research in fault diagnosis training methodology (Johnson, 1981) indicate that performance on TASK, a computer-based fault diagnosis simulation, is somewhat predictive of fault diagnosis, or "troubleshooting", ability. Further investigation of the predictive potential of this simulation was of interest to the Institute of Aviation for several reasons.

Of primary concern was the evaluation of TASK as a predictor of performance in the AMC. At present, admission decisions are based on a combination of personal interviews and a multiple regression equation developed by Trulock (1958). Variables included in the multiple regression equation are American College Test (ACT) scores, high school rank (HSR), and scores on the Survey of Mechanical Insight (SMI). Despite negative reviews of the SMI (Buros, 1974), statistical information compiled by the Institute of Aviation indicates that it is a strong predictor of performance in the AMC.

While the SMI is valuable as a predictor, the possibility of adapting TASK to serve as a computer-based substitute for the SMI was appealing in that the capabilities of the computer could be used for immediate analysis, report, and storage of test results. Applicants could arrange to take the test when most convenient, and advantage could be taken of existing facilities and available personnel for administering the test.

Also contributing to the appeal of TASK for use in testing is the fact that it is context-free, incorporating characteristics common to many mechanical systems rather than representing a particular one. The simple graphic displays assume neither verbal skill nor familiarity with the mechanical depictions upon which many mechanical aptitude tests are based. This aspect of TASK could be important in assembling a test battery for groups of trainees for whom English is a second language, and might eliminate bias toward younger, less experienced students.

Finally, we felt it was reasonable to expect that if TASK could predict performance in the AMC, it might also be generalizable to other areas of technical and vocational training, such as auto maintenance, agricultrual machine maintenance, and, perhaps, electronics.

As a first step in investigating the predictive ability of TASK, the simulation was adapted for testing purposes. Then, a study was conducted to investigate the relationship of performance on TASK to first-semester grade point average (GPA) for four AMC courses.

## THE SIMULATION

TASK was developed by Rouse and colleagues (Rouse, 1979) on a DEC System-10 computer using Techtronix graphics terminals for display. While this configuration was viable for initial implementation and provided the basis for a variety of experimentation (Rouse, 1979; Johnson, 1981; Hunt, 1981; Henneman, 1981), several limitations were inherent in continuing development on that system. Limited access to less than ten terminals in scattered office locations presented problems with student orientation and scheduling in previous studies. Enhancement of the graphic displays and improvements from a human factors perspective were limited by the capacity of the system.

In contrast, the PLATO system, designed specifically for educational applications, offered many features desirable for further improvement and expansion of TASK as a testing instrument. Easily accessible, there are over 600 terminals in various locations throughout campus, ten of them at the Institute of Aviation. In electing to convert TASK to run on the PLATO system, we were able to embed the simulation within a framework that includes an introductory lesson, on-line student assistance, diagnositic feedback, and facilities for the management and monitoring of both student progress and data collection. The graphic displays were made more visually appealing, and the human/machine interface was further optimized with a facility to operate the simulation entirely via touch input. A "hall of fame" displaying the names and scores of the best TASK performers adds an element of competition to the simulation. Pilot tests were conducted to provide formative evaluation as these changes were implemented. The PLATO version of a TASK problem is shown in Figure 1.

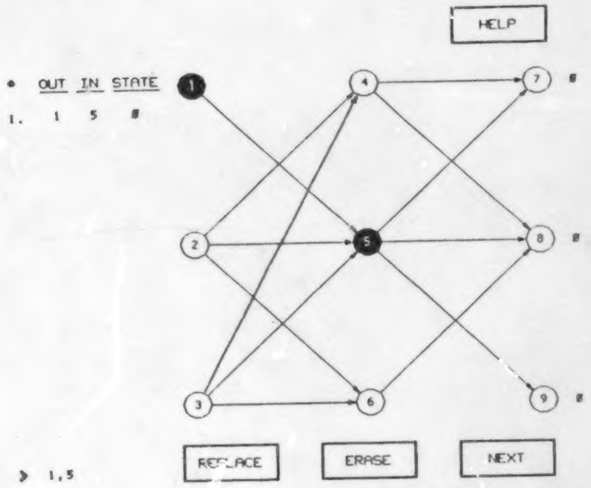


Figure 1

The test consists of a set of problems in which TASK "systems" of varying numbers of components, each having a random number of inputs and outputs, are generated and displayed as in Figure 1. Components produce either a 1, indicating acceptable output, or a 0, indicating unacceptable output. A component will produce a 1 if: 1) all inputs to the component carry acceptable output; and 2) the component has not failed. If either of these conditions is not met, the component will produce a 0. Only one component is randomly selected to fail in each problem.

A problem begins with the display of a system with the outputs of the rightmost components indicated, as shown in Figure 1. Based on this evidence, the subject's task is to "test" connections between components until the failed component is found. In this example, the student touched part number 1 and part number 5 to observe the state of the output. As shown in the upper left, the state of the output from part 1 to part 5 was 0; therefore, part 1 must have failed and must be replaced. A replacement is made by touching the "replace" box and then the part number. If a student needs assistance, the "help" box in the upper right can be touched. TASK problems increase in complexity from a simple 3 x 3 matrix as shown, to 5 x 5 and finally 7 x 7.

PROCEDURE

Twenty-seven first-semester students enrolled in a Powerplant Theory course, Aviation 142, volunteered to participate in this study. It was assumed that this method of selection would introduce no bias. Subjects signed up for one of three contiguous two-hour sessions.

Participants were assigned to PLATO terminals and were given instructions for logging on to the system. Proctors were available to provide assistance throughout the session. Subjects were given an introductory lesson, for which average time to completion was appoximately twenty-five minutes. Then, five practice problems representative of those on the test were presented. No data were collected for these problems.

Subjects then solved twenty preselected TASK problems. Seven of these were 3 x 3 systems, seven were 5 x 5, and six were 7 x 7, presented in random order. Average time to complete this set of problems was approximately forty-five minutes. At the end of the test, subjects were shown their scores and were given the option to look at the TASK "Hall of Fame" before logging off the system.

Four of the twenty-seven participants in this study did not complete the first semester in the AMC. Therefore, data were analyzed for twenty-three subjects.

RESULTS AND DISCUSSION

Data analyzed were the three predictor variables currently in use: ACT scores, HSR, and SMI scores; and an overall TASK score based on 1 unit per observation, 10 units per replacement, and 13 units per minute for each problem solved. The weights assigned to elements in the TASK score were arrived at through pilot testing. Also examined were the individual measures of performance on TASK, including number of observations, number of replacements, and time to solution for each problem.

A correlation analysis was performed to assess the relationship among the predictor variables, TASK performance measures, and GPA. As expected, a significant relationship was found between ACT scores and HSR, ($r = .422$, $p < 0.05$). In that SMI scores did not correlate significantly with either of these measures of cognitive ability, SMI might be considered a measure of aptitude rather than ability. Interesting, though not statistically significant, is the correlation between SMI and TASK, ($r = .386$, $p < 0.10$). Since TASK did not correlate significantly with either ACT or HSR, we might infer from the relationship with SMI that TASK may possibly be a measure of aptitude.

Other correlations significant at the 0.05 level were HSR to GPA, ($r = .438$), and SMI scores to GPA, ($r = .468$). While, as might be expected, average time to solution and average number of replacements per problem were significantly related to TASK score, ($r = .647$ and $.858$, respectively, $p < 0.05$), no other relationships were found to be significant.

Multiple linear regressions were performed to assess the predictive relationships of performance measures to the criterion GPA. Relevant results are summarized in Table 1. Indicated for each regression equation are the predictor variables included, the value of the multiple correlation coefficient ("R"), the percent of variance in the criterion variable accounted for by the predictors ("% Var.), and the probability associated with the variance-ratio ("p"). Raw score regression equations are presented in the lower portion of Table 1.

TABLE 1

Results of Multiple Regression

| Predictors | R | % Var. | p |
|---|---|---|---|
| 1) ACT, HSR | .6384 | 40.76 | .0053 |
| 2) ACT, HSR, SMI | .7136 | 50.92 | .0031 |
| 3) ACT, HSR, TASK | .7017 | 49.24 | .0042 |

Raw score equations:

1) $GPA = 4.1186 - .0693\ ACT + .0214\ HSR$

2) $GPA = 3.8475 - .1631\ ACT + .0182\ HSR + .0080\ SMI$

3) $GPA = 4.9052 - .0675\ ACT + .0206\ HSR + .0193\ TASK$

These results suggest that including either SMI or TASK scores in the multiple regression equation might increase the predictive potential of the equation over use of ACT scores and high school rank alone. (For the addition of SMI, $F(1,19) = 3.9380$, $p < 0.10$; and for TASK, $F(1,19) = 3.1760$, $p < 0.10$.) In further examination of the second and third equations in Table 1, no significant difference was found between SMI and TASK scores in the amount of additional predictive information contributed. While the number of subjects upon which this analysis is based ($n = 23$) is low enough to suggest that some fluctuation in these results can be expected, the similarity in predictive value of TASK and SMI is encouraging.

Though further data would be necessary for confirmation, it seems likely that TASK could be used as a substitute for the SMI as a predictor of performance in the AMC. A follow-up of the progress of participants in this study, and collection of data from subsequent freshman classes will further validate TASK as a testing instrument.

The results of this study suggest that pursuit of related questions is warranted. Currently being planned is an investigation of the relationship of TASK to prevalent standardized mechanical aptitude tests in predicting student performance. The availability of TASK on the PLATO system also provides the basis for studying TASK as a predictor for groups of trainees for whom English is a second language, as well as for students in other technical curricula. Yet to be explored is the role TASK might play in teaching problem solving skills as well as in differentiating problem solving abilities of students of various age groups, particularly at the elementary and secondary levels.

## REFERENCES

Buros, O. K., *The Eighth Mental Measurement Yearbook*, Highland Park, New Jersey, Gryphon Press, 1974.

Henneman, Richard L., *Measures of human performance in fault diagnosis tasks.* Unpublished master's thesis, University of Illinois 1981.

Hunt, R. M., & Rouse, W. B., Problem solving skills of maintenance trainees in diagnosing faults in simulated powerplants. *Human Factors*, 1981, *23*(3), 317-328.

Johnson, W. B., Computer simulations for fault diagnosis training: an empirical study of learning transfer from simulation to live performance (Doctoral dissertaion, University of Illinois). *Dissertation Abstracts International*, 1981, *41* (11) 4625-A. (Universit Microfilms No. 8108555).

Rouse, W. B., Problem solving performance of maintenance trainees in a fault diagnosis task. *Human Factors* 1979, *21*, 195-203.

Trulock, W. D., *Relationships between some student characteristics and performance in the Aircraft Maintenance Curriculum at the University of Illinois.* Unpublished master's thesis, University of Illinois, 1958.

# FACULTY COMPUTER LITERACY: FIVE STRATEGIES

DR. ALAN CREUTZ

CUYAHOGA COMMUNITY COLLEGE

> Perhaps the most fundamental area that requires campus wide discussion and planning is computer or information technology literacy (1)

Whether education is aimed at vocational training, socialization, or intellectual stimulation, computer literacy is today a vital ingredient. Like reading and writing, the task of providing computer literacy to students falls not to the disciplinary specialists, but to the entire faculty.

To accomplish comprehensive computer literacy within a community college faculty combines the problems of introducing technology into a technically unsophisticated group with those of innovation in postsecondary education. The intellectual calibre and curiosity of college faculty eases the introduction of new ideas and methods, but the traditional independence and disciplinary compartmentalization of academics makes any centrally planned strategy difficult to execute.

Dennis defines a three-stage model for faculty literacy (2):

1. Awareness Stage
2. Implementing Stage
3. Maintenance of Growth Stage

The discussion focuses on secondary school teachers and presumes a prior commitment to computing in education.

For most community college faculty it is not enough to provide awareness; community college faculty must be motivated--convinced that the commitment necessary to understand computers will be worth the effort. In a busy schedule of teaching and counselling, many faculty perceive computers as of undoubted value to the initiates but too complex and too esoteric for their own needs. They feel intimidated and threatened, awed and even embarassed.

We have tried a variety of techniques in the Awareness-Motivation Stage, and have had mixed successes. This paper will focus on five different techniques we have tried, and discuss how we have fared.

## 1. Central Policy Decision

The most straight-forward approach to developing computer literacy among the faculty is to declare it a policy, introduce a computing system, and decree its use in the curriculum. In 1977, Cuyahoga Community College, Cleveland, Ohio, determined that Control Data Corporation's PLATO system would become an integral part of the basic skills curriculum of the college. A PLATO lab was installed and faculty instructed in its use in the expectation that they would incorporate it into their pedagogic strategies.

Although not a failure, this approach has not been successful in quite the way envisioned. Faculty feared the computer would replace them or, at least, that their control over teaching would be severely diluted. Few faculty accepted PLATO as a new teaching aid or sought to capitalize on its strengths; most found its weaknesses overwhelming. The inconvenience of a PLATO laboratory, the loss of human contact, the inflexibility of a computer program, the high cost, and deficiencies in its information and pedagogic approach all proved to PLATO's critics that teaching had been denegrated rather than augmented (3).

The unplanned success of PLATO has been in areas tangential to its original purpose at Cuyahoga Community College. Faculty who were not thrust into the PLATO environment were curious and came to explore. A personable and knowledgable PLATO staff eagerly showed novice faculty how to log-on, how to scan the catalog, and how to access any of the thousands of packages on the system. Some faculty found packages to augment their curricula. A few--and not all in the technical fields where you might expect--began to speculate on developing their own programs. We now have faculty creating their own course packages using PLATO's PLM while many of the original target faculty still grouse about PLATO's inadequacies.

## 2. Voluntary Workshops

At the other extreme from a centrally mandated commitment to computer usage is voluntary faculty participation in a sequence of orientation workshops and seminars. We have run a number of well-publicized workshops aimed at various faculty audiences, physically located within the computing center or in faculty conference rooms, dealing with various topics.

Attendance at faculty workshops is usually sparse due to time conflicts, lack of motivation, and the fear of intellectual intimidation. Sessions which are built into an on-going and broader faculty conference series fare better. Attendance is higher, and *ipso facto*, they are more successful. Nevertheless, attendance at the sessions dealing with computers is lower than at companion sessions on other topics.

Faculty usually expect one of two outcomes from workshops:

1. to leave fully able to execute sophisticated tasks on the computer; or
2. to be so lost and confused as to have wasted their time.

Since the results are in between, those with higher expectations are disappointed and, often, alienated, while those expecting confusion are surprised and pleased to learn how simple a computer is. Unfortunately, few with this second orientation even bother to attend. The others rarely recognize that an orientation or introduction is exactly that. Learning how to sign on is fine, but inadequate. Mail systems seem frivolous, and more complex applications seem hopelessly out of reach after a two hour introduction.

## 3. Software Developed for Faculty Needs

If central mandate and voluntary orientation seem to be only moderately successful, how about focusing on user needs? We have actively sought to understand those faculty activities which can be made more efficient or more effective with computer assistance. Faculty are becoming aware of computing through our support for their activities. Commercially available software often fit faculty pedagogic needs. Discussions with faculty may not lead to immediate implementation of computer packages but they do lead to an awareness that computers can be used and that the computing center staff is sensitive to faculty concerns. Once faculty realize that their own college is willing and able to support them, they often come up with their own ideas drawn from their colleagues, their professional literature, or their own creativity.

Discussions with faculty led us to develop and implement software to support on-line testing and the creation hard-copy tests using rather complex question-selection criteria (4). Our CALISS package allows faculty to create their own question banks and control with some precision the format of student on-line tests. The key to CALISS's success is its ease of use and the fact that faculty have complete control over both its content and its dissemination. Thus faculty fully retain the autonomy and creativity which are intrinsic to successful teaching.

We are now embarking on a project to expand the CALISS package into a more sophisticated Classroom Management System which will include assessment testing, prescription generation, and interaction with the on-line student information system (5). From test generation we are moving into tracking and counselling. The record keeping modules of CALISS will be expanded so that data on virtually any type of learning activity (from tutoring sessions to clinical experiences for nursing students) can be included.

From the point of view of developing faculty computer literacy, the success of this package is not only in its ever-expanding usage, but in the high degree of faculty input into its design. Nearly twenty percent of all full-time faculty at Cuyahoga Community

College are now involved in designing the ultimate form of the Classroom Management System. Again, creativity and autonomy are key factors in developing faculty interest.

## 4. Let Them Have Micros - Version I

Microcomputers have been widely touted in education circles and are heavily advertised to the general population as versatile and friendly. Microcomputers would seem a likely vehicle for faculty literacy. Intimidation is minimized and faculty autonomy and creativity can be emphasized.

At the request of a data processing faculty member on one of Cuyahoga Community College's three campuses, the college eagerly set up a microcomputer lab with eight Apple II Plus computers. The hope was that, beyond the clear use by the one faculty member in teaching programming on micros, other faculty would explore the lab as they had done PLATO. Our results have been disappointing.

By having established the lab at the behest of a single faculty member, it has been widely perceived by other faculty to be his lab, designed to his specifications and dedicated to his use. Faculty have assumed, incorrectly, that the hardware/software configuration is specialized. The faculty seem as apathetic towards the micros that they have (but never asked for) as they are toward the mainframe. We have failed to communicate the full potential of the microcomputer lab to the faculty on that campus.

An interesting contrast is the evolving presence of a second microcomputer lab at another Cuyahoga Community College campus. The new lab, created after some delay because of budgetary constraints, is becoming a bone of contention because so many faculty have diverse plans for it. The data processing faculty want to replicate the data processing course. The accounting faculty plan courses using applications and CAI software while the developmental education staff want micros to manage their records. Still others are interested in word processing and electronic communications. The impetus for the second lab came from many different faculty and they see the microcomputers as their tool rather than as a special installation in which they have neither rights nor interest.

## 5. Let Them Have Micros - Version II

Clearly the most successful computer literacy strategy used at Cuyahoga Community College, however, appeals to the child in each faculty member. One of Cuyahoga Community College's most ebullient faculty members, Dr. Sam Spero of the Math Department, runs a faculty literacy project in which each is given an Atari-400 microcomputer for a quarter. The response has been incredible. In just two quarters, nearly fifty faculty and more than a dozen senior college administrators have participated and the waiting line grows daily. Faculty learn a little programming, experiment with graphics and sound generation, try a few commercial software packages and use the Atari as a terminal at home to communicate with the mainframe. They learn electronic communications, mainframe CAI and CMI packages, and explore whatever appeals to them (6).

The virtue of this program in a college-wide faculty literacy program is that it motivates faculty to invest a substantial amount of time and energy into learning about computers. Presumably it is both fun and intellectually stimulating while not being intimidating or structured. It capitalizes on the increasing popularity of microcomputers yet carries no onerous expectation that faculty must learn some set syllabus. Faculty learn what they want, at their own pace, and with minimal scheduling conflict.

Faculty from the Atari project come to the computing center filled with ideas for applications and with enthusiasm. They have a sense of the computer as a tool under their control. Their expectations are realistic and their interest genuine.

## Conclusion

The broadest conclusion to be drawn from these experiences is simple but important. For a new idea in postsecondary education to take hold, the key is the faculty. All the support in the world cannot force an idea onto an apathetic faculty. For real success, faculty creativity and autonomy must be nurtured. With PLATO, the real success has been among faculty who more or less found it themselves. Others saw faculty interest, and they then drifted in. The spread has been by irregular osmosis rather than by orchestrated plan.

Similarly the Atari project succeeds because the faculty member identifies her- or himself and commits a substantial amount of energy to participation. Each controls his/her own activities and sets her/his own goals. The Classroom Management System software development project has been in tune with faculty needs and has drawn on faculty expertise and experience. Faculty are real participants in the creation process and mold the content to their own individual applications.

Prepackaged presentations and prepackaged applications given to the faculty have not been successful in encouraging faculty literacy. Systems adopted by the central administration with little faculty involvement or training programs conceived in the computing center have been far less successful than approaches based on building on faculty concerns.

> Tell me and I forget
> Teach me and I remember
> Involve me and I learn
> Benjamin Franklin

## Bibliography

(1) John W. McCredie, "Campus Information Processing: A New Wave," Educational Record, Fall, 1980.

(2) J. Richard Dennis, "Stages of Development in Introductory Computing to Teachers," Proceedings of the NECC, 1979, June, 1979.

(3) Cuyahoga Community College, Office of Educational Evaluation and Market Research Systems, Phase I PLATO Evaluation: Spring Quarter, 1980, (Cuyahoga Community College: 1980).

(4) Cuyahoga Community College, Office of Systems and Computer Services, "CALISS: Computer Assisted Learning and Instructional Support System--Instructor's Guide," September, 1979.

(5) Cuyahoga Community College, Office of Systems and Computer Services, "CMIS: Specifications for the 1982 Classroom Management Information System Software Development Project," November, 1981.

(6) Sam Spero, "Let Them Have Micros: A Computer Literacy Course for Teachers," Association for Computing Machinery annual meeting, Los Angeles, December, 1981.

# COMPUTER LITERACY FOR TEACHERS:

## A District Do-It-Yourself Kit

Judith A. Hopper, Arapahoe County Schools #6, Littleton, Colorado

Glenn Bruckhart, Arapahoe County Schools #6, Littleton, Colorado

## ABSTRACT

This paper describes techniques and ideas useful in implementing an effective program in teacher computer literacy training. The program, incorporating a three phased approach, allows a school district to introduce computers to its faculty in a low key, nonthreatening way. The use of indistrict personnel as instructors aids in maintaining an emphasis on functional applications and continued computer learning experiences.

## INTRODUCTION

Computers are now an everyday fact of life in business, government and industry. School districts are also commonly using computers for payroll, accounting, inventory, and student schedules and records. In fact, a survey done in 1980 by Al Bork and Jack Chambers found that 90% of the school districts in this country use computers in some way.

But computers still have not had a significant impact on the curriculum in most schools. Teachers are not using them as an instructional tool because they do not know how and often are not aware of their instructional potential. Widespread incorporation of the computer as an instructional tool is certainly a function of the general computer awareness and literacy of teachers. It is therefore essential that training in computer literacy be provided and made attractive and usable to large numbers of staff members in any school district beginning an instructional computing program. Such a training program needs to be face-to-face, non-threatening, functional, and continuing. It is believed this can best be done for any school district by people within that district.

What follows is a description of how one school district in Littleton, Colorado is meeting the challenge of effective teacher literacy training. The program was spearheaded by a small group of teachers who had some background in computing and a strong interest in pioneering such a program.

## PROGRAM DESCRIPTION

The computer literacy program for staff members contained three distinct phases: initial awareness, hands-on introduction, and a computer literacy inservice. These three parts met the district philosophy, i.e. use of computers within all disciplines, and mass student exposure and use of the machines.

For Phase One, initial awareness, a small-scale computer fair was held. A number of local vendors were asked to supply the hardware, relevant software and a representative to answer questions. In addition several in-district teachers agreed to be available to aid in demonstrations and encourage their colleagues to experiment with the machines and courseware.

The entire secondary staff was invited to come on a drop-in basis, walk through, look and interact with the micros on display. The atmosphere was low key, friendly and non-threatening.

The fair proved effective in bringing a large number of teachers in contact with a microcomputer, many for the first time. It also made those teachers aware of at least a few of the possible uses of micros in the classroom.

Each secondary school was asked to

designate one interested staff member as a "computer coach." This person became responsible for the introduction of the computers into his own school and their day-to-day use and maintenance. It therefore was the "computer coach" who implemented Phase Two of the program in his respective school.

The coaches were encouraged to have each department meet with them in the computer room. This afforded the opportunity for each department's members to gain some hands-on experience. Relevant courseware was provided and the department members could discuss how it actually might be used within their curriculum.

The meetings also established the coach as the school's in-house consultant to help them with computer-related problems. The coaches had previous programming skills, catalogs of available software, and enthusiasm to help get teachers started using the computer. They were also urged to identify a pool of competent students to act as computer aides for any teachers desiring classroom assistance.

Phase Three of the literacy program was a one semester hour inservice for all interested secondary staff. The class was held at a centrally located high school and has been offered both during the summer and as an evening course during the school year. Sixteen micros with a networking system were used for the classwork with an alternate model available for demonstrations.

The fifteen class hours were broken into six two and one-half hour sessions. The main goal throughout the six sessions was to help teachers become literate and to be comfortable with the computer. Additional objectives were familiarization of teachers with:

1. Computer vocabulary
2. Existing software
3. Construction of simple programs
4. Capabilities and limitations of computers
5. Computer classroom applications
6. Software sources.

By the end of the course the teachers were able to write some simple programs of their own. They could do some editing of an existing program to make it more useful for their own purposes. They became familiar with some sources of additional software. They were aware of and exposed to both CAI and CMI functions of the machine. And they could see the relationship of the computer to instruction and the part it will undoubtedly be playing in the future.

## EVALUATION

Thus far about 30% of the secondary teachers in the district have completed all three phases. Every time the inservice has been offered the classes have been full to overflowing. Additional advanced programming classes have also been requested. It is anticipated Phase Three will be offered on a continuing basis for as long as there is interest. Ideally all staff should complete the program and work is continuing to get as close to this goal as possible.

As a check on the progress of the participants who have completed the three phases, a follow-up questionnaire was distributed. The compiled results appear in Table I.

Comments accompanying the questionnaire indicated that lack of preparation time was the major factor limiting one's use of the computer. However, virtually all respondents indicated they would like to make greater use of the computer. This indicates the need for additional planning time for initial implementations.

## ANALYSIS

It is also vital that district administrators receive computer literacy training because they approve all requests for the literacy programs themselves as well as the corresponding computer hardware. Therefore, a special five session seminar was held for all of the administrative staff from the assistant superintendents down to the assistant principals.

There are a number of advantages to using district equipment and personnel to get one's own computer revolution started. Teachers instructing teachers from within their own district are less threatening than instruction from an "outside" source. Participants are generally more relaxed in a peer-to-peer relationship and not as reluctant to ask questions and say "I don't understand."

The in-district instructors know the equipment and can be a guiding force through the intricacies of the machine. The participants would actually be learning programming and use of software on their own machines. The knowledge is one-hundred percent transferable.

A low-key approach proved to be a good instructional method. The teachers are there to learn to coexist with and not be threatened by the computer. They

| QUESTION | RESPONSE NEVER | RESPONSE OCCASIONALLY | RESPONSE OFTEN |
|---|---|---|---|
| 1. I have used the computers: | 37% | 50% | 13% |
| 2. I have asked for help from my computer coordinator: | 50% | 41% | 9% |
| 3. I have received help from my computer coordinator: | 38% | 49% | 13% |
| 4. I have used information from the inservice: | 12% | 69% | 19% |
| 5. Since taking the course I am more likely to read articles on computers | 13% | 51% | 36% |
| 6. I would use the computers if I had software readily available: | 6% | 54% | 40% |
| 7. Since taking the course I have found it easier to understand literature containing computer terminology: | 10% | 42% | 48% |

TABLE I. FOLLOW-UP QUESTIONNAIRE RESPONSES

do not have to worry about completing projects, taking midterm exams, studying for finals, or reading required text materials. Although university credit cannot be earned, recertification and salary schedule credit can. Also the course and material covered is structured to one's own district philosophy and goals.

Because the course is taught through the district, it can be tailored to each participant and their particular school needs. Also, a significant advantage to in-district instruction is that the core instructional team is available on a day-to-day basis for problem solving and consultation.

## SUMMARY

In summary, the three-phase in-district program outlined met the four criteria established for such a teacher training program. First, there was face-to-face instruction. Each of the three phases provided the participants with exposure to the machines and someone to answer questions, help, and instruct. Most teachers will neither become nor remain computer users if they must develop the expertise on their own.

Secondly, the situations were non-threatening. For those with no knowledge of computers, the machines can be a mysterious device shrouded in technical jargon. There is a feeling among teachers that only those people trained in mathematics and science have the capabilities required to use a computer effectively. Everything possible was done to dispel this myth and make each participant as comfortable as possible with the machine. The point was made that computers are just that, machines!

Thirdly, the information and instruction given was functional. Throughout the six sessions of the inservice all materials and programs presented were geared to education and how a teacher might make use of the computers with his classes. Problem solving, tutorials, CAI programs, grading software, and simulations were written, discussed and demonstrated with respect to their use in the classroom. Participants needed applications they could take back to their school and classrooms and use.

Finally, the program is continuous. With the instructors in the district, they are available for consulting with those who have completed the program and also to continue the program for as long as it is needed. The inservice phase has now been given seven times for district personnel. Also an advanced level can be added for those desiring more programming instruction and technique.

This literacy program was given partially during the school year. The small scale computer fair was held after school as were the hands-on introductions in the individual schools for each department. The inservice phase was given in a three day summer session and also taught one night a week for six weeks during the school year. The summer sessions were the most productive because the teachers were free from their pedagogical responsibilities and could give their full concentration to the course.

All school districts can implement a similar program. The basic tools needed are: a dedicated team of interested teachers with some computing experience, a cooperative administration, and willing participants. These three tools and the model described above will enable you to "DO IT YOURSELF."

# MICROCOMPUTERS AND ACHIEVEMENT MONITORING

## A Partial Solution to the Demands for Educational Efficiency and Accountability

Charles G. Boody

Hopkins, MN, Public Schools

The past decade has seen a growing demand for educational accountability. Several states, among them Georgia and Florida, have instituted minimum competency requirements, and in all states there is a drive to make education more cost effective; a drive that has been given additional impetus by cutbacks in federal funding for education because of the current economic situation. One sort of criterion-referenced scheme or another would seem in the eyes of many educators, to provide a partial solution to this vexing problem.

Comprehensive Achievement Monitoring (CAM) is one such scheme; one that has been around for a long time. The techniques that have been given the name CAM developed, at least partially, through a 1967 grant from the Charles F. Kettering Foundation to Dwight W. Allen. The earliest developments were undertaken at Stanford University and the University of Massachusetts. The Hopkins, Minnesota, schools have been involved with CAM since the early 1970's. In 1974, their project was granted the National Diffusion Network "Good Housekeeping Seal of Approval" as a project that could be funded for replication. In 1977, Hopkins received funds for dissemination of information about the project. It became evident to Dr. Donald Sension, Director of the Hopkins project, that a major problem for many districts interested in CAM was the need for access to a main-frame computer system. If an interested district did have access, it was unlikely that their main-frame supported a version of CAM, and if the district did not have access, it was even more unlikely that they would be willing to invest in access for one such program, however useful it might be.

For those reasons, efforts were begun in the winter of 1977 to develop a version of CAM on a microcomputer. The Apple II was chosen because of the statewide support in Minnesota given the Apple II through the Minnesota Educational Educational Computing Consortium. I was contacted as a contract programmer, first to investigate the possiblity of undertaking such a project on a micro, and then in the summer of 1978 to write the project. The first version was completed in August of 1978, and since that time it has undergone considerable testing and debugging. It has been running successfully for more than two years now without any known problems, and has been expanded and revised a great deal during that period. A major revision is in progress as this paper is being written, and if it is completed will be discussed at the conference. In the 15 years since the original project, there has been considerable literature published about CAM. Those interested in the detail should read Comprehensive Achievement Monitoring: A Criterion-referenced Evaluation System by Gorth, O'Reilly, and Pinsky (Educational Technology Publications, Englewood Cliffs, NJ, 1975).

To use CAM, one must first define the materials to be taught in terms of objectives. Then, questions designed to test learning of each objective must be written. These must be questions to which the responses can be easily scored by computer: usually multiple choice and/or true-false questions are used. Once this ground work has been completed, the resultant item bank is used to create tests. The data from these tests is then used to provide a variety of information for teachers and students. The specific information in those reports is discussed below.

Fundamental to the CAM philosophy (and also to many discussions of educational methodology) is the idea that proper teaching requires a constant flow of information from pupil to teacher as well as from teacher to pupil. Teachers must know what the student knows about the subject before it is taught and how much has been retained from materials long-since taught if the course is to be well structured. Usually these needs are met by using a "pre-test" before a unit is taught, testing materials from the unit in a unit test, and examining student retention of information through review tests. In the classic CAM model, students are given short tests at regular intervals. Each test (usually about 20 questions) should primarily cover recently-taught information but also should include questions about information to be presented during the next week or two and some questions testing objectives covered some time in the past. Thus, each test serves simultaneously as an indicator of how well students understand materials that have not yet been formally introduced (a "pre-test"), how well students have grasped materials recently taught (a "post-test"), and how well students have retained information taught some time in the past (a "retention test"). The results from such tests provide valuable information for both students and teachers if they are presented in useful ways. In addition to providing a testing model, CAM provides a means of organizing test data results in ways that meet the user's needs.

The student receives a "test coupon" (See Figure 1). For each question in the examination he has just completed, the coupon shows the objective number to which the question refers, indication of whether the question was correctly answered and (using the word YES) indication of whether the question referred to an objective that he had already encountered in the classroom. This makes it easy for the students to refer to remedial materials quickly, and for teachers to pinpoint problems that cause difficulties for specific students. The remainder of the student coupon shows how the student has done on previous tests. Information is shown for all items on a test and also for only those items the student was expected to know. Easy access to this historical information about his work allows the student to monitor his own progress (or lack of it) as the course progresses.

```
************************************************************************
*ANDERSON STEVE                   MATH2                          SMITH *
*   110 TEST PERIOD 4            10/16/81                      FORM 38 *
*             FRACTION CORRECT ON YES ITEMS IS 6/8                     *
*=====================================================================*
*  OBJ A INS   '  OBJ A INS  ' OBJ A INS  '  OBJ A INS  '   OBJ A INS *
*  102 +       '  128 +      '            '             '             *
*  109 + YES   '  128 +      '                                        *
*  110 + YES   '  130 + YES  '                                        *
*  110 + YES   '  130 + YES  '                                        *
*  114 + YES   '  130 - YES  '                                        *
*  118 - YES   '             '                                        *
*  120 -       '             '                                        *
*  120 +       '             '                                        *
*  126 +       '             '                                        *
*  126 +       '                                                      *
*=====================================================================*
*                                                                      *
*          TEST    PCT COR  PCT COR  PCT COR  PCT COR  PCT COR         *
*           ADM     TOTAL     YES      RETN     POST     PRE           *
*            1       88       100      100      100       67           *
*            2       87        90       90        0       80           *
*            3      100       100      100      100      100           *
*            4       80        75       80       67       86           *
*            5                                                         *
*            6                                                         *
*            7                                                         *
*            8                                                         *
*            9                                                         *
*           10                                                         *
*         CUM TOT    89        92       92       91       84           *
************************************************************************
```

FIGURE 1

The teacher can get additional reports that provide information for making basic teaching decisions. A group summary by student provides a sort of "grade book" by repeating the information from the bottom of the student coupon (Figure 2).

```
**********************C.A.M. GROUP SUMMARY BY STUDENT***********************
  COURSE: MATH2                      TEACHER:  SMITH      10/16/1981    PAGE 1
                          TEST ADMINISTRATION PERIOD                      CUM
          1       2       3       4       5       6       7       8       9      10  TOTAL
===========================================================================
  ANDERSON STEVE   ID# 110
    TOT    88      87     100     80                                           89
    YES   100      90     100     75                                           92
   RETN   100      90     100     80                                           92
   POST   100       0     100     67                                           91
    PRE    67      80     100     86                                           84
  CAMPBELL AARON   ID# 120
    TOT   100      87      87     47                                           80
    YES   100      90     100     50                                           86
   RETN   100      90     100     60                                           88
   POST   100       0     100     33                                           82
    PRE   100      80      71     43                                           72
  MALLORY CHRIS  ID# 130
    TOT    38     100      87     47                                           67
    YES    60     100     100     75                                           83
   RETN    50     100     100    100                                           92
   POST    67       0     100     33                                           64
    PRE     0     100      71     14                                           44
```

FIGURE 2

Item analysis is available so that poorly-written test items can be discovered and removed from the item bank (Figure 3).

```
*************C.A.M. SINGLE USE TEST FORM EVALUATION*************
            FORM: 38   COURSE: MATH2   TEACHER:  SMITH
               10/16/81                TEST PERIOD:  4
        ITEM  OBJ  COR  NUM  PER
         NUM  NUM  ANS  RESP COR     A      B      C      D      E     NR
===============================================================
          1   102   2    5    60    40     60      0      0      0      0
          2   109   2    5    80    20     80      0      0      0      0
          3   110   3    5   100     0      0    100      0      0      0
          4   110   1    5   100   100      0      0      0      0      0
          5   114   4    5    40     0     40     20     40      0      0
          6   118   4    5    60     0      0     40     60      0      0
          7   120   2    5    40    60     40      0      0      0      0
          8   120   1    5    20    20     80      0      0      0      0
          9   126   3    5    60    20     20     60      0      0      0
         10   126   2    5    60     0     60     40      0      0      0
         11   128   5    5    40     0      0      0     60     40      0
         12   128   3    5    20    40     20     20     20      0      0
         13   130   1    5    80    80      0     20      0      0      0
         14   130   4    5    60    20     20      0     60      0      0
         15   130   2    5    40     0     40     60      0      0      0

TOTALS                 75    57
```

FIGURE 3

Most important of all, the "group summary by objective" report (Figure 4) allows teachers to see at a glance whether the class knows an objective before it is taught, has learned the recently-taught information, and has retained that knowledge as the term moves along.

```
***********************C.A.M. GROUP SUMMARY BY OBJECTIVE********************
        COURSE: MATH2          TEACHER: SMITH            10/16/81       PAGE 2
                                         TEST ADMINISTRATION PERIOD
                            1      2      3      4      5      6      7      8      9     10
============================================================================================
OBJECTIVE 111
   PCT COR PRE             20     15      0      0      0      0      0      0      0      0
   NUM RSP PRE             20     20      0      0      0      0      0      0      0      0

   PCT COR POST             0      0     96     94      0      0      0      0      0      0
   NUM RSP POST             0      0     20     20      0      0      0      0      0      0

   PCT COR RETN             0      0      0      0     90     88      0      0      0      0
   NUM RSP RETN             0      0      0      0     20     20      0      0      0      0

OBJECTIVE 114
   PCT COR PRE             67      0      0      0      0      0      0      0      0      0
   NUM RSP PRE              6      0      0      0      0      0      0      0      0      0

   PCT COR POST             0      0      0      0      0      0      0      0      0      0
   NUM RSP POST             0      0      0      0      0      0      0      0      0      0

   PCT COR RETN             0     60      0     60      0      0      0      0      0      0
   NUM RSP RETN             0      5      0     10      0      0      0      0      0      0

OBJECTIVE 116
   PCT COR PRE             50      0      0      0      0      0      0      0      0      0
   NUM RSP PRE              6      0      0      0      0      0      0      0      0      0

   PCT COR POST             0      0      0      0      0      0      0      0      0      0
   NUM RSP POST             0      0      0      0      0      0      0      0      0      0

   PCT COR RETN             0    100      0      0      0      0      0      0      0      0
   NUM RSP RETN             0      5      0      0      0      0      0      0      0      0
```

FIGURE 4

The concepts and reports discussed above provide a brief survey of CAM. Hopkins' contribution to the development is in two areas. During the early 1970's, activity was focused around implementing CAM on a district-wide basis using a large timeshare system. These efforts produced a large item bank of objectives and related test questions for many subject matter areas common to elementary and secondary education and a set of procedures that would insure easy access to the item bank and quick response in the production and scoring of tests. It was this work that received the NDN certification. Some 42 books of objectives and associated questions have been developed in the district and have been made available to other districts since 1974. The Evaluation Center staff members have conducted many workshops describing CAM and the Hopkins adaptation of it. No exact records exist; but in recent years, there have been over 30 presentations each year.

It became evident in the process of disseminating information about CAM that a microcomputer-based implementation would make the CAM concepts useful in many more districts.

The Apple version of CAM can handle 200 students divided among up to 4 teachers, 100 objectives, 20 test forms, and 10 testing periods on a single data disk: sufficient for a single course or grade level for one semester in most situations. All of the CAM examples in this paper were produced from the Apple version. Since the program was made publicly available it has been purchased by more than 200 school districts from Florida to Alaska. Clearly there was a need for this sort of monitoring tool.

Such a tool would, of course, be useless if it did not help students learn and teachers teach. That is, though it seems sometimes to be overlooked, what education is all about. CAM, as implemented on the Apple II, does work. A report prepared by the Program Evaluation Center of the Yakima, Washington, Public Schools in June of 1981 states: (See Figure 5)

"An evaluation study was undertaken in the spring of 1981 to determine the impact of the CAM system on student performance. Five classrooms (two at grade four and three at grade six) were identified as having fully utilized the potential of the CAM system. Three classrooms (two at grade four and one at grade six) were selected for comparison. The CAM and non-CAM classrooms at each grade were very similar in terms of such background variables as the proportion of boys to girls, the students' ethnic backgrounds and their socio-economic status, and in terms of the students' scholastic achievement as measured by standardized norm-referenced tests.

"Scores on the district's criterion referenced fourth and sixth grade student learning objectives (SLO) tests in mathematics were used as the indicators of student performance. An analysis of covariance was employed to analyze the scores, with the total battery scores from the California Achievement Test at grade four, and from the Stanford Achievement Test at grade six, as the covariants.

"The analysis demonstrated that the performance of CAM students at both grades four and six was superior to that of their non-CAM counterparts.

"....Not only are the mean scores from CAM classrooms significantly higher than those from non-CAM classrooms, but the scores of CAM students are more tightly clustered; the standard deviations for the CAM groups are just half as large as those for the non-CAM groups. The superior performance of, and greater homogeneity of achievement among, CAM students indicated the effectiveness of the CAM system."

During the past three years, two things have become clear about using CAM on the Apple. First, as Figure 5 clearly indicates, the adaptation of CAM to the microcomputer environment has not decreased its effectiveness as an aid to the teaching-learning process. Secondly, having a micro-based system moves CAM closer to the teachers and, in most cases, decreases the time needed to score the test and return it to the student. Teachers appreciate the responsiveness of the system, and the more direct access to it. Somehow the whole concept seems less threatening to them when they have more direct control of the computer. These two things have lead the Hopkins schools to begin replacing the well-established timeshare CAM system with the Apple-based system. Initial efforts appear to be going very well though no hard data is available at this point. One can easily envision a future time when each teacher will have access to CAM-like programs in his or her own classroom.

| | | CAT#<br>SAT## | SLO###<br>SCORES |
|---|---|---|---|
| GRADE 4 | Non-CAM (N=20) MEAN | 442.98 | 53.34 |
| | STD DEV | 32.75 | 5.46 |
| | CAM (N=20) MEAN | 444.36 | 56.44* |
| | STD DEV | 33.11 | 2.60 |
| GRADE 6 | Non-CAM (N=41) MEAN | 328.65 | 32.55 |
| | STD DEV | 90.65 | 13.14 |
| | CAM (N=41) MEAN | 327.20 | 52.30** |
| | STD DEV | 88.16 | 7.26 |

# Grade 4
## Grade 6
\###Mathematics

*f=11.9 @ 1/79 df, favoring CAM students at the .01 level of confidence

**f=53.4 @ 1/37 df, favoring CAM students at the .01 level of confidence

FIGURE 5

The effectiveness of the Apple-based CAM program has lead Hopkins to develop two other Apple-based programs for monitoring student progress. The first, called HEI-Score is a quick and simple test scoring package. Teachers can enter a test-key card and then the response cards. The program scores the responses and provides student scores indication of how well the student did on each question, a test form item analysis, and the distribution of the scores with indication of mean, median, and standard deviation (See Figure 6).

```
                           STUDENT SCORES                PAGE 1
                                             KEY: MATH 12 10-23-81
I.D.            #CRT %CRT ABCBDECDDA
---------------------------------------------------------------------
   123456       8/10  80  ..x.x.....
JOHN SMITH      7/10  70  .x....xx..
JOE             4/10  40  xx.x.oox..
333 33 222      9/10  90  ...x......
ANY INFORMATION 9/10  90  ...0......


                    MATH12 10-23-81 TEST ANALYSIS
QUEST         NUMBER OF RESPONSES (TOT.RESP.=5)    PERCENTAGE RESPONSES
          A      B      C      D      E     NR   A    B    C    D    E     NR
-------------------------------------------------------------------------------
1         4*     0      1      0      0     0 ' 80*   0   20    0    0     0
2         2      3*     0      0      0     0 ' 40   60*   0    0    0     0
3         1      0      4*     0      0     0 ' 20    0   80*   0    0     0
4         0      2*     0      2      0     1 '  0   40*   0   40    0    20
5         0      0      1      4*     0     0 '  0    0   20   80*   0     0
6         0      0      0      0      4*    1 '  0    0    0    0   80*   20
7         0      0      3*     1      0     1 '  0    0   60*  20    0    20
8         1      3*     1      0      0     0 ' 20   60*  20    0    0     0
9         0      5*     0      0      0     0 '  0  100*   0    0    0     0
10        5*     0      0      0      0     0 '100    0    0    0    0     0


            MATH12 10-23-81 FREQUENCY DISTRIBUTION (10=PERFECT SCORE)
RAW   FREQ FREQ                                              CUM CUM
SCORE  CNT  %                 DISTRIBUTION                   CNT  %
-----------------------------------------------------------------------
  9     2  40 ****************************************    '  2   40
+ 8     1  20 ********************                        '  3   60
@ 7     1  20 ********************                        '  4   80
  6     0   0                                             '  4   80
  5     0   0                                             '  4   80
  4     1  20 ********************                        '  5  100
=======================================================================
PERCENTAGES:            1         2         3         4
                        0         0         0         0

NUMBER OF STUDENTS: 5
@ IS MEAN SCORE:    7.4
+ IS MEDIAN SCORE:  8
STANDARD DEVIATION: 2.07
```

<u>FIGURE 6</u>

The second, called Mastery Management, allows teachers to prescribe levels of mastery (as a percentage of correct responses to items testing each objective) and keeps track of mastery information on each student for each objective. Many states and/or individual schools are moving toward a minimum competency approach to education, and are in need of a management tool to help them. A single Mastery Management disk can manage 900 students associated with up to 20 teachers, 400 objectives, and 50 test forms (interchangeable at any time). The student receives a coupon similar to the CAM coupon, and the teacher can get a variety of reports. A typical report is shown in Figure 7.

```
*******************TEACHER OBJECTIVE REPORT 11/18/81******************
TEACHER: JANE FELTL    MATH2 OBJECTIVES 102 TO 128        NOT MASTERED
                                                          INFORMATION
OBJECTIVE:
# 102 ADD-20             # 110 USE + OR -          # 118 GROUP ADDENDS
# 104 SUBTRACT-18        # 111 DIFFERENCE          # 119 SUB,TWO DIGIT #
# 105 ADDEND AND SUM     # 112 MISSING PARTS       # 120 RELATION-ADD&S
# 106 MISSING ADDEND     # 113 ADD 3 NUMBERS       # 121 ADD-CARRYING
# 107 USE PROPER WORD    # 114 ADD 2 ADDENDS       # 122 SUB-BORROWING
# 108 COMMUNATIVE        # 116 ADD 3 ADDENDS       # 124 2-STEP EQUATION
# 109 ZERO CONCEPT       # 117 CHECK PROBLEMS      # 128 EQUIVALENT ?

PAGE 1                      1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                            0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 2 2 2 2 2
                            2 4 5 6 7 8 9 0 1 2 3 4 6 7 8 9 0 1 2 4 8
ID# STUDENT          TIES#
=====================================================================
900 DANIEL AGRIMSON  234897 . . . . . . . . . . . . . . . . . . . . .
902 STEVEN BECK      789253 . . . . . . . . . . . . . . . . . N . . .
904 TERRY CARR       369823 . . . . N . . . . . . . . N N . . N . N .
906 DOYLE CORNETT    675841 . . N . N . N . . . . . . N N N N N N N N
908 JO GALLAGHER     404344 . . . . . . . . . . . . . . . N . N . N N
910 BOB GRIMES       270873 . . . . . . . . . . . . . . . . . N N N .
```

FIGURE 7

All three of these programs support entry using card readers or page readers, and Mastery Management and CAM support entry from the computer keyboard as well. Our experience has been very positive with the small, inexpensive card reader made by HEI, inc. and HEI-Score is written specifically for that reader. Mastery Management also supports that reader. CAM supports the HEI reader as well as card readers from Chatsworth and the Scantron page reader.

All three of these systems can use the same response cards so that the teachers, should they choose to do so, get a wide variety of different kinds of useful information efficiently. Currently, a system is under development that will integrate the features of all three of these programs into a single very flexible package with additional capabilities of summing across data diskettes to provide district-wide information. The results of those efforts are not available as this paper is being written.

In summary then, the work done at Hopkins indicates that: CAM provides a useful and valid tool for improving classroom instruction, CAM and similar programs sufficiently sophisticated to be very valuable, can be implemented successfully on microcomputers, and finally that microcomputer-based implementations can speed turn-around time for teachers and help teachers overcome distrust of computers by making it possible for the teachers to process their own test data. The resultant data can, in turn, provide evidence for the success of a school district and, in the hands of a good teacher, can enhance the learning of students. Clearly, this approach provides a partial answer to the intertwined problems of efficiency and accountablility in education.

# THE AFFECTIVE RESPONSE OF STUDENTS TO CAI

William E. Russell

MERRIMACK COLLEGE

## ABSTRACT

The primary objective of this paper is to consider the affective response of students to the use of the computer as an educational tool. Student feelings about the tasks demanded in coursework seem crucial to the successful accomplishment of important cognitive objectives. As a required course component, all students in two courses of the Education curriculum completed during the semester a series of CAI lessons. At the end of the course, a semantic differential was used to record student feelings toward this experience in computer-assisted-learning. This paper reports the results as being mildly positive.

In educational literature, three divisions or dimensions of human behavior are often distinguished: the cognitive, the psychomotor, and the affective. The affective focuses upon emotional or feeling responses to an experience. The primary objective of this paper is to consider the affective response of students to the use of the computer as an instructional tool. An instructor intends to design experiences for students that are not only educational but meaningful and positive. The variables learning and liking are mutually reinforcing. The assumption (perhaps a fact) is that in any course student feelings about the tasks demanded are crucial if cognitive objectives, generally the chief responsibility of the teacher, are to be achieved. While the ability to understand and process information may be the ultimate goal of a course, actual achievement may depend upon perceptions, attitudes, feelings, satisfactions, dissatisfactions and the values of the student - that is, upon a student's affective response. Experience in the first instance is affective. Later, the content can be analyzed and interpreted. The affective response of students to the instructional use of the computer seemed significant as a factor of study.

## Course Application

For the past three semesters, I have integrated the computer into the syllabus of two of my courses - The American School and Foundations of Education. The programs are conversational, that is, interactive written in an authoring language packaged by Digital Equipment Corporation. The author language, termed DECAL (Digital Equipment CAI Author Language) enables the writer to create lessons consisting of up to 50 segments.

For my courses, I have written two types of lessons each with a specific purpose. One type allows the student to review interactively and individually the main concepts of the class text. After reading each chapter, the student can test his understanding of the content by running the appropriate computer program. Each chapter of the required text has a corresponding computer lesson. The typical lesson has fifteen segments, requiring approximately 15-20 minutes of student time. The second lesson type is directly instructional in that no prior student preparation is necessary. The purpose here is not a review or self-check, but formal instruction. New concepts are presented with opportunities for application and analysis. These lessons are longer, consisting of an average of 35 segments, and each demands about 35-50 minutes of student effort.

For most students, this was the first personal encounter with the computer as a teaching device. Initial yet typical difficulties were experienced such as problems with loggin, typing errors, and misplaced identification codes. After a week, these minor frustrations evaporated. From then on, the system seemed to function smoothly. I became interested, however, in the student perception or attitude toward the computer as a teaching device. The charge has been made of impersonalization, mechanization, and dehumanization. Programmers apparently feel sensitive to this concern, thus the penchant to "humanize" - "Greetings," "I've been off the job for a day," "Hurry back," "I'll see you at lesson 4," "Good work." On another tack, three fourths of my students are in a teaching certification program. If computers are to achieve their potential as an instructional aide, teachers must be reached. Innovations in Education enter ploddingly. Computers seem to be their own advocate, after exposure that is if the experience is pleasant as well as challenging. It seemed important if one teaches as one's taught to employ methods deemed rich, individualized, and human then to elicit feedback on student perception of the medium. Students perceive as receivers, teachers as senders. Thus, it is vital for the instructor to discover student perception of method.

## Measurement of AFFECT

As stated at the outset, this paper primarily concerns the affective response of students to the use of the computer. The success of an instructional program depends to some extent on the positive affective response of students. Learning and liking may go hand in hand. Scales can be constructed to measure the intangibles of the affective domain, though the results are hardly exact. One technique developed by Charles Osgood is the semantic differential, a bipolar adjective scale.* Usually, this bipolar adjective scale consists of a seven point scale linking one adjective with its opposite. The adjective pair describes an experience or objects in this case the use of computers in my two courses. The main task of the evaluator is to select appropriate adjectives. Evaluative type terms are particularly apt in measuring attitudes. In addition, adjective pairs can be grouped so that subscales are produced, each intended to assess a particular factor. Thus, the semantic differential can be considered a collection of subscales. As a matter of form, not all positive terms, for example, should appear on the same side. The student placement indicates direction as well as intensity of reaction. The semantic differential constructed is produced below with directions preceeding:

Place an X in that one space of the seven between each adjective pair that best indicates your perception of the use of computers in this course. The closer you place your X toward one adjective or the other, the more you think that adjective better describes the use of the computer in this course than the other.

1 2 3 4 5 6 7

unpleasant _:_:_:_:_:_:_:pleasant
original _:_:_:_:_:_:_:conventional
creative _:_:_:_:_:_:_:routinized
cautious _:_:_:_:_:_:_:experimenting
unfriendly _:_:_:_:_:_:_:sociable
beautiful _:_:_:_:_:_:_:ugly
passive _:_:_:_:_:_:_:active
hostile _:_:_:_:_:_:_:amiable
timid _:_:_:_:_:_:_:adventurous
gentle _:_:_:_:_:_:_:harsh
slow _:_:_:_:_:_:_:fast
pleasing _:_:_:_:_:_:_:annoying
lively _:_:_:_:_:_:_:tedious
bad _:_:_:_:_:_:_:good
encouraging _:_:_:_:_:_:_:superfluous
monologic _:_:_:_:_:_:_:dialogic
meaningful _:_:_:_:_:_:_:superfluous
sharp _:_:_:_:_:_:_:dull
boring _:_:_:_:_:_:_:satisfying
relevant _:_:_:_:_:_:_:irrelevant
enhancing _:_:_:_:_:_:_:diminishing
wasteful _:_:_:_:_:_:_:effective
involved _:_:_:_:_:_:_:distant
dull _:_:_:_:_:_:_:interesting
beneficial _:_:_:_:_:_:_:nonconsequential

## Results of Semantic Differential

To present the results in more graphic form, each adjectival pair is set in the assumed positive direction, i.e., the more positive term appears to the right. The average placement of the 77 student respondents is represented by the asterisk at the correct location on the 1 to 7 continuum. The precise average appears to the right.

|  | 1 2 3 4 5 6 7 |  | AVG. |
|---|---|---|---|
| unpleasant | _:_:_:_*_:_:_ | pleasant | 4.82 |
| conventional | _:_:_:*:_:_:_ | original | 4.61 |
| routinized | _:_:_:*:_:_:_ | creative | 4.25 |
| cautious | _:_:_:_:*:_:_ | experimenting | 5.44 |
| unfriendly | _:_:_:*:_:_:_ | sociable | 4.64 |
| ugly | _:_:*:_:_:_:_ | beautiful | 3.06 |
| passive | _:_:_:_:*:_:_ | active | 5.29 |
| hostile | _:_:_:_*_:_:_ | amiable | 4.83 |
| timid | _:_:_:_:*:_:_ | adventurous | 5.23 |
| harsh | _:_*_:_:_:_:_ | gentle | 2.87 |
| slow | _:_:_:_:*:_:_ | fast | 5.46 |
| annoying | _:_:*:_:_:_:_ | pleasing | 3.17 |
| tedious | _:_:_*_:_:_:_ | lively | 3.43 |
| bad | _:_:_:_:*:_:_ | good | 5.35 |

---

*Osgood, C., Suci, ., & Tannenbaum, P. - The Measurement of Meaning - Urbana, Ill. University of Illinois Press 1967

Results of Semantic Differential cont.

| | 1 2 3 4 5 6 7 | | AVG. |
|---|---|---|---|
| discouraging | _:_:_*:_:_:_:_ | :encouraging | 3.69 |
| monologic | _:_:_:*:_:_:_ | :dialogic | 4.19 |
| superfluous | _:_:_:*:_:_:_ | :meaningful | 4.17 |
| dull | _:_:_:*:_:_:_ | :sharp | 3.89 |
| boring | _:_:_:_:*:_:_ | :satisfying | 5.04 |
| irrelevant | _:_:_:_*:_:_:_ | :relevant | 4.67 |
| diminishing | _:_:_:*:_:_:_ | :enhancing | 4.17 |
| wasteful | _:_:_:_:_*:_:_ | :effective | 5.53 |
| distant | _:_:_:*:_:_:_ | :involved | 4.17 |
| dull | _:_:_:_:_*:_:_ | :interesting | 5.36 |
| nonconsequential | _:_:_:_*:_:_:_ | :beneficial | 4.68 |

The direction and intensity of the response can thus be judged to see the overall pattern. Items most strongly endorsed, positively as well as negatively, can be identified. From an assumed positive viewpoint, students considered the descriptors "experimenting," "active," "adventurous," "fast," "good," "satisfying," "effective," and "interesting," to apply more intensely to their experiences with these CAI lessons. Descriptors "harsh," "ugly," "annoying," "tedious," "discouraging," on the negative side, received relatively strong endorsements. The table following presents the items ranked in order of the average response from most positive to most negative.

| | 1 2 3 4 5 6 7 | | AVG. |
|---|---|---|---|
| wasteful | _:_:_:_:_*:_:_ | :effective | 5.53 |
| slow | _:_:_:_:_*:_:_ | :fast | 5.46 |
| cautious | _:_:_:_:_*:_:_ | :experimenting | 5.44 |
| dull | _:_:_:_:_*:_:_ | :interesting | 5.36 |
| bad | _:_:_:_:_*:_:_ | :good | 5.35 |
| passive | _:_:_:_:_*:_:_ | :active | 5.29 |
| timid | _:_:_:_:*:_:_ | :adventurous | 5.23 |
| boring | _:_:_:_:*:_:_ | :satisfying | 5.04 |
| hostile | _:_:_:_:*:_:_ | :amiable | 4.83 |
| unpleasant | _:_:_:_:*:_:_ | :pleasant | 4.82 |
| nonconsequential | _:_:_:_*:_:_:_ | :beneficial | 4.68 |
| irrelevant | _:_:_:_*:_:_:_ | :relevant | 4.67 |
| unfriendly | _:_:_:_*:_:_:_ | :sociable | 4.64 |
| conventional | _:_:_:_*:_:_:_ | :original | 4.61 |
| routinized | _:_:_:_*:_:_:_ | :creative | 4.25 |
| monologic | _:_:_:*:_:_:_ | :dialogic | 4.19 |
| superfluous | _:_:_:*:_:_:_ | :meaningful | 4.17 |
| diminishing | _:_:_:*:_:_:_ | :enhancing | 4.17 |
| distant | _:_:_:*:_:_:_ | :involved | 4.17 |
| dull | _:_:_:*:_:_:_ | :sharp | 3.89 |
| discouraging | _:_:_*:_:_:_:_ | :encouraging | 3.69 |
| tedious | _:_:_*:_:_:_:_ | :lively | 3.43 |
| annoying | _:_:*:_:_:_:_ | :pleasing | 3.17 |
| ugly | _:_:*:_:_:_:_ | :beautiful | 3.06 |
| harsh | _:_:*:_:_:_:_ | :gentle | 2.87 |

Each item can be considered part of a group. The groupings because of the terms chosen seem natural rather than imposed or designed. Thus, the individual scales of which there are twenty-five can form clusters in which each single item can be viewed as the same or similar, from the viewpoint of the person judging, as other items in that cluster. For example, three groupings could be made since a relationship of similarity exists between items such as (1) "experimenting," "interesting," "adventurous," (2) "active," "fast," and (3) "good," "satisfying," and "effective." The first three related terms while they might be considered an expression of the students' initial personal encounter with a new technology, may also deserve a more significant interpretation. Each term may connote that CAI has a built-in motivational feature that appeals forcefully to students. If true, this gives CAI an advantage to be exploited. The second grouping confirms the teacher perception that CAI in contrast to the traditional method of instruction engages the student. The claim that CAI provides an "interactive environment" seems warranted. The third related grouping - "good," "satisfying," "effective," - are descriptions as well as evaluations of the student perception of their experience with CAI lessons. As quite general terms, they may summarize many positive feelings of the students toward the experience. This gives some evidence that CAI accomplishes a teaching goal. Grouping these similar items into seemingly natural clusters or factors produces three tables. Each cluster is named for convenience and description.

Originality

| | 1 2 3 4 5 6 7 | | AVG. |
|---|---|---|---|
| conventional | _:_:_:_*:_:_:_ | :original | 4.61 |
| routinized | _:_:_:*:_:_:_ | :creative | 4.25 |
| cautious | _:_:_:_:_*:_:_ | :experimenting | 5.44 |
| timid | _:_:_:_:*:_:_ | :adventurous | 5.23 |
| dull | _:_:_:_:_*:_:_ | :interesting | 5.36 |
| Average of 5 items | | | 4.98 |

Dynamism

| | 1 2 3 4 5 6 7 | | AVG. |
|---|---|---|---|
| passive | _:_:_:_:_*:_:_ | :active | 5.29 |
| slow | _:_:_:_:_*:_:_ | :fast | 5.46 |
| distant | _:_:_:*:_:_:_ | :involved | 4.17 |
| tedious | _:_:_*:_:_:_:_ | :lively | 3.45 |
| Average of 4 items | | | 4.59 |

cont.

Satisfaction

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | AVG. |
|---|---|---|---|---|---|---|---|---|---|
| unpleasant | | | | | * | | | pleasant | 4.82 |
| unfriendly | | | | | * | | | sociable | 4.64 |
| ugly | | | * | | | | | beautiful | 3.06 |
| hostile | | | | | * | | | amiable | 4.83 |
| annoying | | | * | | | | | pleasing | 3.17 |
| bad | | | | | * | | | good | 5.35 |
| boring | | | | | * | | | satisfying | 5.04 |
| Average of 7 items | | | | | | | | | 4.40 |

On the negative side, descriptors such as "harsh," "ugly," "annoying," "tedious," and "discouraging," appear. While interpretation is difficult relying simply upon the endorsements, student verbal complaints during the semester may furnish a clue. For example, the most frequently voiced objections to the CAI lesson was "I knew the answer but typed (or spelled) the word incorrectly. The computer told me I was wrong."

Checking the lesson reports did verify the frequency of spelling errors. The developers of DECAL have devised a partial remedy to this limitation by allowing the lesson writer to insert an asterisk in key word responses. The asterisk will automatically match any characters that appear in the student answer between the characters surrounding the asterisk in the writer's version. This is no doubt a concession to poor spellers, but may remedy some of the negative perceptions. During the semester also some students felt that the lengthier lessons required a sustained concentration allowing for no pause or relief. The terminal continued for 35 or more minutes uninterruptedly to type text or questions. In fact, one more inventive student told me he simply turned off the power to get a moment's respite. Because responses were timed, the lesson proceeded inexorably to the consternation of some students. If too deliberate, the student could only watch as the lesson moved on to the next segment after typing a comment such as "Remember, you only have 30 seconds to respond." If these factors contributed at least in part to the more negative responses, they are rather easily remedied. In any case, the negative endorsements deserve in themselves a more thorough investigation if the goal is to make CAI a more effective learning technique.

It certainly seems more useful to look at items individually and in natural clusters. However, it was also intended in design that the twenty-five items of this semantic differential be a collection of three sub-scales. One sub-scale attempts to assess student evaluation of their experiences at the computer terminal. The second sub-scale elicits student attitude toward the computer as a learning vehicle for the particular content of the course, i.e., the cognitive goals. The third sub-scale assesses student attitude toward the computer as a teaching tool. Each sub-scale is presented below with the results.

Evaluation of the Experience

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | AVG. |
|---|---|---|---|---|---|---|---|---|---|
| unpleasant | | | | | * | | | pleasant | 4.82 |
| unfriendly | | | | | * | | | sociable | 4.64 |
| ugly | | | * | | | | | beautiful | 3.06 |
| hostile | | | | | * | | | amiable | 4.83 |
| gentle | | | * | | | | | harsh | 2.87 |
| annoying | | | * | | | | | pleasing | 3.17 |
| tedious | | | * | | | | | lively | 3.43 |
| bad | | | | | * | | | good | 5.35 |
| discouraging | | | | * | | | | encouraging | 3.69 |
| boring | | | | | * | | | satisfying | 5.04 |
| dull | | | | | * | | | interesting | 5.36 |
| Average of 11 items | | | | | | | | | 4.21 |

Attitude Toward Computer
As A Learning Vehicle

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | AVG. |
|---|---|---|---|---|---|---|---|---|---|
| superfluous | | | | * | | | | meaningful | 4.[illegible] |
| irrelevant | | | | | * | | | relevant | 4.67 |
| diminishing | | | | * | | | | enhancing | 4.17 |
| wasteful | | | | | * | | | effective | 5.53 |
| nonconsequential | | | | | * | | | beneficial | 4.68 |
| Average of 5 items | | | | | | | | | 4.60 |

Attitude Toward Computer
As a Teaching Tool

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | AVG. |
|---|---|---|---|---|---|---|---|---|---|
| conventional | | | | | * | | | original | 4.61 |
| routinized | | | | * | | | | creative | 4.25 |
| cautious | | | | | * | | | experimenting | 5.44 |
| passive | | | | | * | | | active | 5.29 |
| timid | | | | | * | | | adventurous | 5.23 |
| slow | | | | | * | | | fast | 5.46 |
| monologic | | | | * | | | | dialogic | 4.19 |
| dull | | | | * | | | | sharp | 3.89 |
| distant | | | | * | | | | involved | 4.17 |
| Average of 9 items | | | | | | | | | 4.73 |

As can be seen, each sub-scale does show a mildly positive response.

It should be stated that an educational method is not easily separated from the content. In this case, the quality of the written programs enhances or detracts from the evaluation of affect. As the programs are improved, one could predict an improvement in affect. The study indicates the present state of the CAI lessons. Also, the semantic differential does not yield exact and easily interpreted results. The approach is semi-projective in that the student is not asked direct questions. This procedure, however, adds to the difficulty of confident interpretation. Nevertheless, the rather large number of items and the consistency of direction toward the positive pole, though the intensity in some cases is not very pronounced, provides useful feedback. This is particularly true if the programs are improved with remedies found to eliminate unnecessary frustrations. This first study, then, can function as a baseline for future measurement of affect concerning the use of the computer in my courses. Furthermore, the overall pattern of student response encourages this neophyte in CAI lesson writing to pursue the method by improving existing programs and creating a larger and more varied library of lessons.

# THE COMPUTER AS A COMPONENT IN A MULTI-METHOD APPROACH TO TEACHING ANATOMY AND PHYSIOLOGY

Donald A. Wheeler - Cuyahoga Community College

Mary Jane Burge - Cuyahoga Community College

Abstract: Computers have been incorporated into a multi-method approach to teaching anatomy and physiology at the community college level. The computer is integrated with the following methods of instruction:

1. lecture booklet of performance objectives
2. lecture workbook of drawings
3. color-coded lecture outline overlays
4. laboratory manuals
5. slide/tape/script book review package
6. encased, keyed models
7. labeled photo print files
8. films and videotapes

Computer technology has allowed the development of the following three programs:

1. self-tutorial question routine
2. direct selection of objective statements routine
3. anatomical terminology spelling drill and practice

## INTRODUCTION

A multi-method approach to the instruction of anatomy and physiology has been organized for use at Cuyahoga Community College. Anatomy and physiology at our institution is a three-quarter sequence. Each of the courses is divided into a lecture portion and a laboratory portion.

## LECTURE

OVERVIEW: For the lecture, a workbook has been developed consisting of a detailed set of specific student performance (level III) objectives plus a booklet of unlabeled, but "letter-tagged" diagrams. A brief description of our terminology may be helpful at this point. Level I objectives consist of the catalog description and general objectives of the course. Level II objectives are the topical outline for the course put into objective form. Level III objectives describe exactly what the student must do (list, describe, define, state, etc.) for each level II objective. The lecture workbook also contains a packet of unlabeled, "letter-tagged," drawings which the students fill out in lecture as the instructor covers this material. This accomplishes the following:

1. It saves student note-taking time and, therefore, lecture time.
2. More complex drawings can be used than those which could be drawn in lecture.
3. It is a learning experience for the student to seek out the correct structure and label it as it is being discussed by the instructor. This allows for the visual development of complex concepts. Conceptualization of complex material is a problem for many of our students.

As lecture instructors, we have developed a complete set of overhead transparencies which consist of a color-coded outline of the subject material coordinated with the performance objectives. Each level I objective is developed into a module. Each level II objective is numbered on the overhead transparency corresponding to the performance objective booklet. Under each level II objective on the transparency is the information in outline form, printed in contrasting color, needed to answer every performance objective in the student workbook. Incorporated within the outline are transparencies with labeled diagrams which correspond to the unlabeled diagrams in the student workbook. As instructors, we use the drawings during lecture to develop certain complex concepts. At the same time, the students follow the lecture by labeling their unlabeled diagrams in their workbooks.

The process of labeling and note-taking on the diagrams allows the student to visually construct a concept from which to study at a later date. After each lecture, the student should be able to use his/her lecture notes to complete the performance objectives for the material covered. If a student can not answer a particular performance objective or section of objectives, then the computer can be his/her resource for the information as described below.

COMPUTER APPLICATION: We are presently working on a computer-assisted learning project based on our performance objectives. The student has two options when he/she signs on to the computer. The first is what we call the self-tutorial question routine, and the second is the direct selection of objective statements routine. In the first routine, the student is able to select which unit of material and which level II objective (with its performance objectives) that he/she would like to be tested on. The student is presented with a sequential presentation of questions. A correct response continues the presentation. If the response is incorrect, the computer presents on the screen a statement which incorporates the answer to the question. The student is told to read the statement and then try again to answer the question.

Once the student has finished this routine, the computer informs the student how many questions he/she answered correctly out of the total number of questions presented, along with the percent score. The student's score is then compared to a predetermined "success score" and then a recommendation is made as to whether or not he/she needs to acquire a better knowledge of the material before taking the lecture exam.

The instructor has access to a computer file with the following data for each student:

1. name and I.D. number of the student.
2. unit material used by the student.
3. number of correct responses.
4. total number of questions asked.
5. percent score.
6. date of exercise.
7. time of day.
8. amount of time spent on the exercise.
9. cumulative time usage throughout the quarter.

The second routine, direct selection of objective statements, allows the student to directly select the appropriate performance objectives in statement form. This provides the student with the information necessary to complete the performance objectives in his/her workbook. This would be of benefit to both those students who have missed the lecture entirely, and to those who attended the lecture but have found that, upon trying to complete the objectives in their lecture workbook, they were missing some of the information.

LABORATORY

The laboratory material is presented with the aid of overhead transparencies, 2x2 slides, models, and specimens. The material is presented by the instructor in outline form. For this we use overhead transparencies which include the outline, plus diagrams that are exact copies of the unlabeled, but letter-tagged, drawings from the lab manuals. These diagrams were produced by our own artist directly from the models used in the lab. As the instructor is presenting the material from the transparencies, the student follows along and labels the same diagram in his/her manual.

The lab manuals have been written in the following way for the following purposes:

1. At the beginning of each lab unit is a listing of level II objectives which are the main topics to be discussed during that lab.
2. The main body of each lab unit consists of introductory information, including vocabulary terms, experiments, and so on, in addition to unlabeled, letter-tagged diagrams with an alphabetical list of structures to be located on the diagram. The purpose of these unlabeled diagrams is to give the student an exercise in locating and labeling the structures that are presently being presented in class by the instructor. This will enable the student to form a visual relationship of the various components within the anatomical structure being discussed.

Led by the instructor, the class as a whole then proceeds to identify the previously labeled diagram structures on 2x2 slides of models, specimens, microscope slides, and so on. Once this point has been reached, the students are then free to investigate, at their own tables, the models, specimens, microscope slides, and so forth, for all the structures previously labeled on the drawings and found on 2x2 slides.

3. The last part of each lab unit is a set of performance objectives which list and describe each of the points of information that must be mastered by the students in order to have a full command of the material. These performance objectives specify exactly what the student is responsible for locating on each of the models, specimens, microscope slides, and so on, as well as performance objectives relating to the functions of various anatomical structures and physiological processes. This information is presented by the instructor during his/her presentation of the material.

## TESTING OF LABORATORY INFORMATION

Testing of the laboratory information is accomplished mainly by the use of 2x2 slides which the instructor uses to ask questions on both structure and function. Blank slides are inserted at various points to allow for non-slide related questions about the material.

## SELF-TUTORIAL STUDY ROOM

A self-tutorial study room has been established to provide the student with additional study time. This learning resource room is available to students on an average of 40 hours per week. With this flexibility, students can organize their study time around their academic and personal schedules. This flexibility is a necessity for the student population at CCC: parents with family responsibilities, working students, and many part-time students.

The following resources are available in the study room:

1. A complete reference source has been organized in order to meet the performance objectives for lab. This is accomplished by having at least one of every model used in lab on display at a designated station. Each model is tagged and encased in plexiglass. The plexiglass encasement ensures that the identification tags remain in their correct position and also prevents damage to the models, some of which are quite fragile. Preservation of the models ensures a constant, accurate reference source for the student.

2. For those objectives which require microscope slides, diagrams, or specimens, photo print files consisting of tagged and labeled 8x10 photo prints have been placed at a resource station in the study room. These prints have been made directly from the 2x2 slides found in the students' self-tutorial packets.

3. Self-tutorial packets containing carousels of 2x2 slides, audio tapes, and script books have been produced in order to facilitate learning through two of the cognitive learning modes, visual and auditory. The 2x2 slides found in the packet are the same ones used in lab, and the same as those used in the testing process. The 2x2 slides have been produced so that every anatomical structure that the student is held responsible for can be seen on a slide. The audio tapes instruct the student as to what he/she should be able to locate on each 2x2 slide. Another important function of the tape is to present the correct pronunciation of the terminology. This will help the student not only with his/her own pronunciation but also with spelling, which has been a critical area of concern in the past with our students.

The script book lists visually each of the stuctures to be located by the student on each 2x2 slide in order to fulfill the performance objectives in the lab manual. The student is instructed via audio tapes and script books to locate various structures on each unlabeled slide. A student unable to locate any of the listed structures is directed to a station containing the tagged and keyed model shown on that slide. At this station, he/she can find the information necessary to complete that objective.

4. A PLATO computer terminal is located in this room which allows the student to participate in the Self-Tutorial Question Routine and Direct Selection of Objective Statements Routine. Sixteen other PLATO terminals are located in the Learning Resource Center for student usage.

5. An ATARI 800 microprocesser computer system has been set up with the following:

a. Hear and Spell (An Anatomical Terminology Spelling Drill and Practice Program). This system incorporates an ATARI tape recorder controlled by the computer. The computer turns on the tape recorder, which then pronounces the word to be spelled. Then the tape is turned off, and the computer waits for the student to type in the word correctly. If the student types in the correctly spelled word, then the computer will turn on the tape automatically to give the next word. If the student misses the word, then the computer will ask the student if he/she wants to try the word again or compare his/her answer with the correct spelling of the word before trying the word a second time. If the student chooses to make the comparison, the computer will present his/her response with the correctly spelled word. His/her incorrectly spelled word appears in large blue letters at the top of the screen. The correctly spelled word initially appears in a flashing mode, then is displayed continuously in large yellow letters in the middle of the screen. All of this appears on a bright red background. After the comparison is made, the student is then instructed to try the word again.

b. Self-Testing Objective Review. This program is similar to the Hear and Spell program described above. It allows the student to review the lecture objectives in a testing format. The computer turns the tape recorder on, which presents a question testing the chosen objectives. The program allows questions requiring one to four answers. If the student responds incorrectly, he/she has the option either to try again immediately or to see the answer(s) before making another attempt. If he/she chooses to see a display of the correct answer(s), the answer(s) will be presented in color graphics similar to the Hear and Spell program described above. The program will accept responses in singular or plural forms, as well as responses containing extra words. Multiple answers can be entered in any order.

6. Tutorial assistance is provided by a resource person who not only aids the student with individual questions but also manages the room and its equipment. In addition, there is also a room attendant to check the learning packets in and out, and carry out other student management functions.

## ANALYSIS OF COMPUTER USAGE

An analysis using proportions was computed for the total group. Ninety percent of the objectives were successfully met by students of the group who self selectively elected to use the computer program, whereas only 82% of the objectives were met by students who did not use the computer program. The difference was found to be significant using the test for significance of differences between the two proportions. A Z value of 4.0 was obtained, significant at the 1.98 level.

## FUNDING

This project was made possible through the generous funding obtained under the AIDP and SDIP grants. These grants allowed for the purchase of models and equipment, and faculty release time.

# THE USE OF PRODUCTION RULES IN THE DEVELOPMENT OF AN EXPERT TUTOR FOR ALGEBRA

WILLIAM S. BREGAR
Oregon State University

ARTHUR M. FARLEY
University of Oregon

BRIAN LANTZ
Rutgers University

## ABSTRACT

Production systems, sets of rules and an accompanying interpreter, have been employed by researchers in Artificial Intelligence to produce expert systems. In this paper we describe how this technique can be utilized in a CAI environment for teaching students to solve linear algebraic equations. We present two experimental versions of such a system. One is capable of solving and explaining how it solved a problem. The other interacts with a student who can command the system to perform the usual set of algebraic manipulations. Suggestions, following the same lines of reasoning as the student, can be presented if the student cannot proceed or is seen to be going in the wrong direction. Dialogues and samples of rules in the system are given to illustrate how the problem solving knowledge can be made explicit and utilized.

## INTRODUCTION

Intelligent computer-assisted instruction (ICAI) has become an increasingly important domain of research in computer-based instruction. A variety of systems in several domains have been developed in recent years. Among these are SOPHIE (Brown and Burton, 1975) for teaching electronic troubleshooting, WHY (Bates, Brown, and Collins, 1979) a tutor for geographical concepts and GUIDON, (Clancey, 1979) a system for teaching medical problem solving. While most such systems are experimental, they provide a basis for significantly improving the tutorial component of a CAI system. The major reason for this increase in tutorial capabilities in these systems is the emphasis placed on the representation of domain and problem solving knowledge and the subsequent uses of that representation.

The purpose of this paper is to describe one aspect of the representation of knowledge in an ICAI system for teaching problem solving for high school algebra word problems. In particular we will discuss the concept of a rule-based system and demonstrate how it can be used to provide tutorial guidance to a student. We will base our discussion on two versions of a system for tutoring a student attempting to solve a set of linear equations which could have been derived from a problem statement. A more general view of our system is described in Bregar and Farley (1980) and a theory of knowledge representation for the interpretation of word problems and conversion into equations is given in Bregar and Farley (in preparation).

## Version I - A Demonstration of Algebra Problem Solving

Dialogue 1 depicts a session with the first version of the algebra tutor. The student interacts with the system to establish a goal variable and upon further questioning, provides it with an equation containing the goal. The system manipulates the input until it has isolated the goal on the left hand side and reduced its coefficient to 1. If there are still unknowns on the right hand side, the system establishes them as subgoals and requests information as before. It continues in this fashion until it can compute a numeric result for an unknown. It then propagates this result back through its list of subgoals and associated equations, substituting and simplifying until the main goal has been solved.

The system proceeds to summarize

Dialogue 1

```
TO BEGIN, PLEASE GIVE ME A LIST OF THE VARIABLES IN THE PROBLEM.
TYPE THEM IN ON 1 LINE SEPARATED BY COMMAS.
LISTENING:  X,Y,Z  (Items after LISTENING:  are input by student).

OK!  WHICH VARIABLE MUST WE SOLVE FOR?
LISTENING:  Y

SO Y IS OUR GOAL VARIABLE.

PLEASE INPUT NEW INFORMATION ABOUT Y.
TYPE IN AN EQUATION CONTAINING THE GOAL.
LISTENING:  4Y - (3/2) Z = 4X-6

OK!  WE HAVE NEW INFORMATION ABOUT Y.

THE EQUATION IS NOT SIMPLIFIED IN TERMS OF THE GOAL.
SO WE SIMPLIFY IT.

WE ISOLATE THE GOAL VARIABLE ON THE LEFT SIDE
THIS PRODUCES THE EQUATION 4Y = 4X - 6 + (3/2)Z.

WE REDUCE THE GOAL TERM MAKING THE COEFFICIENT EQUAL TO ONE.
THIS PRODUCES THE EQUATION Y = X - (3/2) + (3/8)Z.

TYPE IN RETURN TO CONTINUE (PSOLVE periodically halts to allow
                            students to inspect actions).
WE NOTICE THAT THE VARIABLE X APPEARS
IN THE EQUATION Y = X - (3/2) + (3/8)Z

SO WE WILL CONSIDER X TO BE OUR NEW GOAL.
WE MUST SOLVE FOR IT IN ORDER TO SOLVE FOR Y.
WE WILL REMEMBER Y AND GO BACK TO IT LATER.

PLEASE INPUT NEW INFORMATION ABOUT X,
TYPE IN AN EQUATION CONTAINING THE GOAL.
LISTENING:  X = 5Z - (3/4)

OK!  WE HAVE NEW INFORMATION ABOUT X.

WE NOTICE THAT THE VARIABLE Z APPEARS
IN THE EQUATION X = 5Z - (3/4)

SO WE WILL CONSIDER Z TO BE OUR NEW GOAL.
WE MUST SOLVE FOR IT IN ORDER TO SOLVE FOR X.
WE WILL PUT THAT VARIABLE TO THE SIDE AND SOLVE FOR IT LATER.

PLEASE TYPE IN NEW INFORMATION ABOUT Z,
TYPE IN AN EQUATION CONTAINING THE GOAL.
LISTENING:  Z = 3

OK!  WE HAVE NEW INFORMATION ABOUT Z.

NOW WE HAVE A RESULT.
Z IS EQUAL TO 3.

SO WE RETURN TO THE PARENT GOAL VARIABLE WHICH IS X.
```

Dialogue 1

## Dialogue 1 (Continued)

```
WE CAN SUBSTITUTE 3 for Z IN X = 5Z - (3/4).
THE EQUATION IS NOW X = 15 - (3/4).
WHICH RESULTS IN THE FOLLOWING EQUATION X = (57/4).

TYPE IN RETURN TO CONTINUE

AHA!  WE HAVE DETERMINED A RESULT.
X IS EQUAL TO (57/4).

SO WE RETURN TO THE PARENT GOAL VARIABLE, WHICH IS Y.

WE CAN SUBSTITUTE (57/4) for X IN Y = X - (3/2) + (3/8)Z
THE EQUATION IS NOW Y = (57/4) - (3/2) + (3/8)Z

THE EQUATION IS NOT SIMPLIFIED IN TERMS OF THE GOAL.
SO WE SIMPLIFY IT.

WE COMBINE TERMS FROM THE RIGHT SIDE INVOLVING CONSTANTS.
WHICH RESULTS IN THE FOLLOWING EQUATION Y = (51/4) = (3/8)Z

TYPE IN RETURN TO CONTINUE

WE REMEMBER THAT Z IS EQUAL TO 3.
WE CAN SUBSTITUTE 3 FOR Z IN Y = (51/4) + (3/8)Z.
THE EQUATION IS NOW Y = (51/4) + (9/8)

THE EQUATION IS NOT SIMPLIFIED IN TERMS OF THE GOAL.
SO WE SIMPLIFY IT.

WE CAN COMBINE TERMS FROM THE RIGHT SIDE INVOLVING CONSTANTS.
THE RESULTING EQUATION IS Y = (111/8).

TYPE IN RETURN TO CONTINUE

NOW WE HAVE A RESULT.
Y IS EQUAL TO (111/8).

OUR FINAL RESULT IS
Y EQUALS (111/8) or 13 and 7/8.

      SUMMARY

WE BEGAN AS ALWAYS BY GETTING A LIST OF THE VARIABLES
WHICH WERE X,Y,Z AND IDENTIFIED THE MAIN GOAL AS Y.

WE OBTAINED AN EQUATION ABOUT Y,
WHICH WAS 4Y - (3/2)Z = 4X - 6.

THIS WAS SIMPLIFIED TO OBTAIN Y = X - (3/2) = (3/8)Z.

WE NOTICED THE OCCURRENCE OF X.

WE OBTAINED AN EQUATION ABOUT X,
WHICH WAS X = 5Z - (3/4).

WE NOTICED THE OCCURRENCE OF Z.

WE OBTAINED AN EQUATION ABOUT Z,
WHICH WAS Z = 3.

WE NOTICED THE RESULT THAT Z EQUALED 3.
```

Dialogue 1

Dialogue 1 (Continued)

```
WE THEN RETURNED TO OUR PREVIOUS GOAL WHICH WAS C,
RECALLING THE EQUATION:  X = 52 - (3/4).

WE SUBSTITUTED THE 3 for Z,
WHICH WAS SIMPLIFIED TO OBTAIN X = (57/4).

WE NOTICED THE RESULT THAT X EQUALED (57/4).

WE THEN RETURNED TO OUR PREVIOUS GOAL WHICH WAS Y
RECALLING EQUATION Y = X - (3/2) + (3/8)Z.

WE SUBSTITUTED THE (57/4) FOR X,
WHICH, AFTER SIMPLIFICATION YEILDED Y = (51/4) = (3/8)Z.

WE REMEMBERED THAT Z WAS EQUAL TO 3.

WE SUBSTITUTED THE 3 FOR Z
TO OBTAIN THE EQUATION:  Y = (51/4) + (9/8).

THIS WAS SIMPLIFIED TO OBTAIN Y = (111/8).

SO WE OBTAINED THE FINAL RESULT THAT Y = (111/8) or 13 and 7/8.


HERE IS A NEW PROBLEM FOR YOU TO SOLVE.
THE PROBLEM IS SIMILAR TO THE ONE WE JUST SOLVED.
YOU ARE GIVEN THE FOLLOWING EQUATIONS:

(4/3)E - 2 = (4/3)U - (1/2)C
(5/3)C - (1/4) = (1/3)E
12 = 4C

YOU MUST NOW SOLVE FOR THE VARIABLE U.
```

Dialogue 1

Example Productions

1. (SUBSTITUTE
   (if (there is a variable V1 marked 'GOAL' in M) and (there is a string marked 'INFO' in M which states a relation between the goal and another variable V2) and (there is a string marked 'RESULT' which states an equality between V2 and a constant C) then (replace V2 with C in the string marked 'INFO')))

2. (NOTE-VAR-OCCURRENCE-MAKE-GOAL
   (if (there is a variable V1 marked 'GOAL' in M) and (there is a string S1 marked 'INFO' in M which contains the goal) and (there is a variable V2 in S1 which is not the goal) then (mark V2 'GOAL') and (mark V1 'PARENT' of V2)))

3. (RETURN-TO-PARENT-GOAL
   (if (there is a variable V1 marked 'GOAL' in M) and (there is a string marked 'RESULT' in M which sets V1 equal to a result R1) and (there is a variable V2 marked 'PARENT of V1 in M) then (make V2 the current goal)))

4. (COMBINE-CONSTANTS
   (if (there is a variable V1 marked 'GOAL' in M) and (there is a string S1 marked 'INFO' in M which contains G1) and (there is a sub string S2 in S1 of the form <constant> $\{\begin{smallmatrix}+\\-\\\times\end{smallmatrix}\}$ <constant>) then (replace S2 with the result of the operation applied to the constants)))

5. (COMBINE-TERMS-ON-LEFT-SIDE
   (if (there is a string EQN marked 'INFO' in M) and (there are strings T1, T2, ..., TN on the left side of EQN which have the same variable) then (add T1, T2, ..., TN and replace them with the result on the left hand side of EQN)))

6. (MOVE-GOAL-TO-LEFT-SIDE
   (if (there is a variable V1 marked 'GOAL' in M) and (there is a string EQN marked 'INFO' in M) and (V1 appears on the right hand side of EQN as part of a term T1) then (negate the coefficient part of T1 and add T1 to both sides of EQN)))

7. (REDUCE-GOAL-TERM
   (if (there is a variable V1 marked 'GOAL' in M) and (there is a string EQN marked 'INFO' in M containing V1 as part of a term T1) and (the coefficient part of T1 is not equal to 1) then (divide both sides of EQN by the coefficient part of T1)))

Figure 1. Sample productions in a system for solving linear equations. M stands for Memory. Not all productions are shown here, nor are they in the same form they take in the actual system. Note that some productions are concerned with strategy (establishing goals and processing them), others with algebraic operations.

the steps it took to solve the problem, and then, poses a similar problem to the student. Of particular note is the explicitness with which each step is presented, and the "granularity" or the degree of detail of the presentation. Another aspect of this system is the level of expertise displayed in solving the given problem. The system attacks a problem in a top-down fashion, identifying goals and subgoals to be solved in a logical progression of problem solving behaviors. This constitutes a means-ends strategy (Newell and Simon, 1972) in which subgoals are generated which reflect differences between a goal state and the current state. Here, the goal is to obtain a value for a given goal variable and differences are represented as intermediate unknowns for which values must be found. The "means" are the set of operations available for "reducing" or eliminating differences. In this example these are the applicable algebraic operations which can be performed on a given equation.

One final aspect of this example, which we will emphasize further later, is the ability for the system to explicate what it is doing. This ability is a direct result of the representation of problem solving knowledge. Each aspect of the problem solving process is described to the system in the form of a conditional statement called a production rule. An individual rule consists of a condition part and an action part and can be interpreted as saying

if <condition> is true then do <action>.

A set of such rules interacting with a memory component is called a production system (PS). The memory typically contains a description of the current problem state including, what is known, what remains to be solved and what the current goal being processed is. The action part of a rule typically modifies the memory component.

A PS operates as a sequence of cycles. In a cycle each rule in the set is examined until one whose condition part matches some portion of memory is found, whereupon its action part is invoked. The rule is thus said to fire completing the cycle. This match-fire sequence is then repeated until either no rule can fire or a rule's action part explicitly halts the system. A production system monitor (PSMON) controls the selection, matching and firing of a rule, starting at the beginning of the rule set in each cycle.

Rather than firing the first production rule whose condition part is satisfied, the PSMON can continue through the rules, collecting all those whose condition parts are matched in memory, and then, according to arbitrary criteria, select one for firing. Such a set of rules is called a conflict set and the procedure for selecting a single rule from this set is referred to as a conflict resolution. A complete description of various methods for handling conflict resolution can be found in McDermott and Forgy (1978). Operating in this fashion, PS's have been made to exhibit complex adaptive behavior in a variety of problem solving situations.

It can now be seen how it is possible to interact with a student user as in Dialogue 1. Figure 1 contains an English like translation of several of the productions used in solving the given problem. Each production has a name indicative of its function and represents a specific "chunk" of knowledge necessary to either perform the required algebraic manipulations (A-rules) or to perform some type of strategic processing (S-rules). All the rules in the system are not given here, however, it should be noted that for the first version of the system, the first matched rule is the one which fires, thus, the ordering of the rules represents a de facto strategy for processing which can be inferred from the dialogue.

As information is acquired about a goal or subgoals, a sequence of productions fires corresponding to the sequence of operations necessary to isolate the designated variable on the left hand side of the equation with a coefficient of 1. The sequence of productions and the associated values assigned to variables in their condition parts as a result of matching the memory is saved and used as input to a routine which performs a summary of the problem solving activity demonstrated to the student.

## Version II. - Adding an Interactive Tutorial Component

In its first incarnation, the system could demonstrate the acquisition of information and processing of that information toward solving a set of linear equations. The only interaction between the student and the system occurred at the initiation of the system which requested information until it had obtained enough to solve the problem.

Version II of the system includes several enhancements as can be seen from dialogue 2. Here, after collecting information about problem variables, the student is requested to provide direc-

## Dialogue 2

```
?(tutor algebra)
(TYPE IN THE VARIABLES FOR YOUR PROBLEM)
?(x y)

(OK WHICH VARIABLE MUST WE SOLVE FOR?)
?(y)

(SO y IS OUR GOAL VARIABLE)

(PLEASE INPUT NEW INFORMATION ABOUT y)
(TYPE IN AN EQUATION CONTAINING THE GOAL)
?8y-3x=14-x+5

(OK!  WE HAVE NEW INFORMATION ABOUT y)

(THE EQUATION IS NOT SIMPLIFIED IN TERMS OF THE GOAL)
(SO YOU NEED TO SIMPLIFY IT)
(WHAT DO YOU WANT TO DO)
?(move the y term to the left side)

(MOVING THE GOAL TERM TO THE LEFT SIDE)
(PRODUCES THE EQUATION (8y-3x-y=14-x+5)
(WHAT NEXT?)
?(isolate the goal term)

(ISOLATING THE GOAL VARIABLE ON THE LEFT SIDE)
(PRODUCES THE EQUATION (8y=14-x+5+3x+y))
(WHAT WOULD YOU LIKE TO DO NEXT?)
?(subtract 8y from both sides)

(COMBINING TERMS FROM OPPOSITE SIDES INVOLVING y)
(PRODUCES THE EQUATION (0=-7y+14-x+5+3x))
(ENTER YOUR NEXT INSTRUCTION)
(combine the x terms)

(COMBINING TERMS FROM THE RIGHT SIDE INVOLVING x)
(PRODUCES THE EQUATION (0=2x+19-7y))
(WHAT NOW)
?(divide everything by 7)

(YOU NEED TO DO SOMETHING ELSE BEFORE YOU DO THAT)
(WHAT DO YOU WANT TO DO)
?(help)

(TRY DOING SOMETHING INVOLVING y)
?(help)

(I WILL DO THE NEXT STEP FOR YOU)
(MOVING THE GOAL TERM TO THE LEFT SIDE)
(PRODUCES THE EQUATION (7y=2x+19))
(WHAT NEXT)
?(divide by 7)

(MAKING THE COEFFICIENT OF THE GOAL TERM EQUAL TO 1)
(PRODUCES THE EQUATION (y=2/7x+19/7)
```

Dialogue 2

tives to the system for isolating the goal variable. A student who is stuck may ask for help, otherwise, he/she may enter a natural language command corresponding to any of a set of normal arithmetic operations or algebraic manipulations. If the command can be interpreted, the system enters its tutorial mode. First, it compares the student's command to the set of operations which could legally be applied to the problem in its current state. If any of these has been selected, the system performs it and displays the result. The legal set of operations, in fact, is the set of possible production rules whose condition parts match the current contents of memory - e.g., the conflict set. The particular production selected is based on how well a keyword analysis of the user's input text compares to production rule names, and variable names (and a dictionary of their synonyms) associated with each rule in the conflict set.

When no rule in the conflict set is indicated by the user's input, the system works ahead an arbitrary number of steps from the current problem state to see if some intervening steps have been skipped. All possible paths are explored to the specified distance. If, in fact, the user specified step appears at this point, the system requests the intermediate steps from the student to insure that pure guesswork was not involved and to reinforce the procedures presumably utilized. If the student is unable to provide the appropriate inputs, the system performs the steps and displays each as it goes. On the other hand, if the system works ahead and finds no correlation to what the student has entered, it will request an alternative operation. When the student is at a loss for what to do, he/she can ask for help and receive a hint. In the event the hint is insufficient, the system will demonstrate its idea of what step to do next. Both the hint and the demonstration are based on the system's selection of a production from the currently applicable conflict set.

Finally, if the student appears to be hopelessly confused and the system cannot generate a solution from the current state, it can back up to a previous state and ask the student to enter a different command.

## DISCUSSION

The two versions of the described system represent what can be called a "knowledge-based" approach to CAI. The production rules comprising the problem solving knowledge in the system make explicit both what is being done and how it is accomplished. Combining a history of rule firings with the current state of the memory component provides the system with an individualized student model. Recall that when, in Version II, the system looks ahead to evaluate a student response provide guidance, its starting point is the current problem state. Thus, the analysis is based on an extension of the logic used by the student and the system to attain that state, not simply on how the system would do the problem on its own.

Because of the interpretive nature of the implementation of a production system, it is not difficult to examine sequences of rules which have fired, undo sequences to return to previous states, or to try several alternatives in an attempt to understand and provide guidance to a student in a tutorial session. All associations made between variables in the rules and the contents of memory for a matched rule are also available for providing hints at varying levels of detail. Since the rules are stored in symbolic form with their names and variable designations intact, the keyword processing of student inputs is also facilitated. Straightforward conversions of instantiated rules (rules with variables assigned to memory elements) are made to generate summaries for the student. The granularity of the rules and the independent chunks of knowledge they represent allow the system designer to experiment with the level of detail with which to treat the problem domain, as well as to modify the system easily with a minimum of impact on the rest of the system.

Finally, the execution of the system results in a natural sequence of problem solving steps. It is driven by a means-ends strategy, although, other strategies could easily be added and employed. It would be difficult to approximate this level of behavior programming the system in an ordinary algorithmic language.

## CONCLUSION

We have demonstrated the utility of production systems, an artificial intelligence tool, as a basis for intelligent, tutorial CAI. Although it is still experimental, our system has the potential for tutoring students who have difficulty solving linear algebraic equations by engaging them in an interactive problem solving episode.

## References

Bates, M., Brown, G. and Collins, A. Socratic teaching of casual knowledge and reasoning. Proceedings of the 1979 ADCIS Conference, San Diego, February, 1979.

Bregar, W.S. and Farley, A.M. Artificial intelligence approaches to computer-based instruction. JCBI, 1980, 6 (4), 106-114.

Bregar, W.S. and Farley, A.M. Knowledge representation in an automated algebra tutor. In preparation.

Brown, J.S. and Burton, R. Multiple representations of knowledge for tutorial reasoning. In D. G. Bobrow and A. Collins (Eds.), Representation and Understanding. New York: Academic Press, 1975, 311-349.

Clancey, W.J. Transfer of rule-based expertise through a tutorial dialogue. PhD Thesis. Computer Science Department, Stanford University, August, 1979.

Newell, A. and Simon, H.A. Human Problem Solving. Englewood Cliffs, N.J. Prentice-Hall, 1972.

# MICROCOMPUTERS & THE HANDICAPPED

CHRIS DUMPER

Technical aids, designed to help the physically handicapped have in the past tended to be cumbersome, expensive and extremely rigid in terms of the benefits they provided and the method by which the user accessed the device. A scanning typewriter for instance was just that - a scanning typewriter - and use of this aid demanded physical manipulation of a microswitch. Similarly, an aid designed to offer its user environmental control such as the turning off and on of radios, lights and alarms, offered solely this and did not have the plasticity of offering anything else.

The introduction of the microcomputer as a technical aid has changed all this. In the first place, as far as aids go, a microcomputer is relatively inexpensive. Equipped with disk drive and a letter-quality printer, it'll probably set you back approximately $4,000.00. Before I continue however, let me state that, as I see it, the prime areas wherein the handicapped may benefit from the aid of electrical or mechanical assistance are communication, environmental control, education and recreation. To many, "recreation" in terms of microcomputers, brings on images of space wars, fights against evil necromancers and exotic adventures--and is often frowned upon when introduced as "education". However, if the purpose of education is to prepare a student so that he can cope effectively with life--then it's as important to teach someone whose future implies an institution or group home independent recreation as it is important to teach your ablebodied student the skills he will require in the work force. The flexibility inherent in the computer allows it to offer the handicapped independence in all these areas. Furthermore, they may be purchased and serviced locally.

Adaptation of the computer so that it may serve as an aid may be accomplished in two very divergent ways. One way is to write appropriate software or "programs". The other is to design hardware so that the disabled have access to any of the programs that can be purchased at your local dealer.

Some of the adapted software we have found very useful in the classroom are two "communication" programs. One, entitled "Handicapped Typewriter" allows, through microswitch manipulation of a cursor, the typing of messages and letters, use of scanning calculator, environmental control, and telephone dialing through use of a "modem". Whereas the old "mechanical" scanning typewriter demanded that the user scan for every letter that he uses in his message, this program features a "file" system whereby the user can store words, phrases and sentences that he finds he frequently uses. By drawing these out of the files and inserting them into the message he's generating he can speed up his written output. The second program was developed at the Trace Centre, University of Wisconsin and allows, through a variety of input mechanisms, the printing of BLISS SYMBOLIC messages. This program is obviously extremely exciting to any BLISS user as he or she can now communicate with individuals not immediately present. By using an additional piece of hardware and an appropriate interface the program will also speak out the messages using a computer generated voice. Although the voice is R2D2-like, it's a lot of fun for someone who is non-verbal.

Adapting games to a scanning-type of input system allows even the severely handicapped student to independently play games and not just your concrete checker and chess type of game but also the more abstract such as Dungeon and Dragon and Wilderness campaign. These program adaptations are not difficult to obtain. Simply go to your local computer dealer and peruse through the programs he is bound to have in a "contributed library". As long as you select programs written in BASIC and that do not require paddle manipulation, your grade 11 or 12 high-school student studying computer science will be capable of reprogramming the inputs required to a scanning system that involves microswitch connection to the paddle 0 or 1 buttons.

Specially written as well as adapted software has obviously been extremely beneficial in offering a source of flexible communication and environmental control to the severely handicapped and non-verbal individual. Just one other program that I'd like to make mention of before rambling on is one that offers written output through a scanning system. This particular program however, is dynamic in that, as you generate a word, it will attempt to predict what your next required letter will be. It then proceeds to position this letter as close to the scanning cursor as possible. It manages to do this by having the many rules of grammar and syntax that apply to the English language programmed into its structure.

Adapting the hardware: In other words designing equipment that attaches to the microcomputer in order that it can be used by the handicapped is another approach that has had much success. For the blind and the deaf, this has included voice output and interfacing to a number of different Braile devices. With respect to physical handicaps this approach has had great success in providing alternatives to the standard keyboard. Key-guards and expanded keyboards are now available for individuals with limited control of their arms and fingers. For those with even less physical control there is available a scanning, or by means of touch or joystick. In this way even the severely disabled can make use of the same programs, such as word and data processors, games and educational courseware that are offered the able-bodied through computer dealerships and school systems.

The greatest benefit that I feel will be offered by this approach is that it will give the handicapped, and especially those confined to group homes and institutions, use of Teledon. Although Teledon is only in its infancy it will eventually allow for shopping, banking, community awareness and education directly from your livingroom. By using the keyboard alternatives mentioned earlier, this "homebound" population will be offered considerable independence.

The field of microcomputer adaptation as a technical aid to be used in the home and school has evolved extremely rapidly with much being accomplished in a trial and error, adhock manner. These however are the growing pains that must be experienced by someone in order that others in the future can benefit. What I have attempted to describe covers the extent to which we have managed to make use of this new technology in our school setting and would not have been possible without a lot of pioneering efforts by private individuals and service groups and funds supplied by them, industry and B. C. Provincial I.Y.D.P.

To give you a glimpse of where this technology may lead in the future, I'd like to briefly outline some projects I had the fortune to see at a recent conference on "Personal Computing and the Handicapped" organized through John Hopkins University.

## HEARING SPEECH AND LANGUAGE

Row/Column Scanning with a Dynamic Matrix, R. L. Jones: This uses a row/column scanning technique however it attempts to predict the likelyhood of the following letter in in the message being generated and displays these letters proximal to the location of the cursor.

Telephone Communication for the Deaf, R. E. Glass: Whereas in the past a deaf person could only communicate over the telephone through teletype devices, the Message Converter will now allow him to communicate with anyone who has access to a simple touch tone telephone.

Think it Through: An Interactive Videodisc for the Hearing Impaired, C. G. Nugent: By mating an MCA DiscoVision 7820 videodisc player to a TRS-80 microcomputer these researchers have managed to introduced sign language to standard CAI programs.

Adapting the Apple II for Physically Handicapped Users: Two Different Solutions: P. Schwejda and J. McDonald: While having produced software applicable particularly to your low functioning physically handicapped person, they also have available in firmware system operable through a number of different input devices, that allows the access to off-the-shelf software.

Computer Assisted Instruction in Expressive and Receptive Fingerspelling, A. J. Welch, T. G. Mancuso, L. P. Licata: Formatted into Learning, Testing and Receptive modes, this program, once paired using high resolution graphics to the TRS-80, may be used to instruct its user in finger spelling.

## LEARNING DISABILITIES AND MENTAL RETARDATION

Computer Assisted Instruction in Special Education, R. P. Swenson and J. C. Kingman: An evaluation of the use of CAI programs with the mentally retarded and the development of subject matter intrinsically motivating to this population.

Microcomputers in Special Education, D. K. Crowther: Programs written to teach basic counting skills and concepts to retarded students operating at a three to seven year old level.

## MOVEMENT, NEUROMUSCULAR AND NEUROLOGICAL

The Motor-Handicapped Support System, R. O. Launey III: Through hardware which adds to the computer voice recognition, sound synthesis, a BSR home control interface and two RS232 communications ports the user can type letters by simply dictating the letters to the computer. Someone without the use of their upper extremities can also exert environmental control or, by mating the system to an automated microfiche reader, he can call up the required book, turn pages or scan--all through verbal commands. (Note: I have seen this system operate with C.P. individuals with extremely laboured speech).

Computer Programs for Cognitive Rehabilitation: Personal Computing for Survivors of brain injury, R. Gianutsos and C. Klitner: Programs which provide rehabilitation to survivors of strokes or head injuries to recover visual perception and memory.

The Eyetracker Communication System, M. B. Friedman, G. Kiliany, M. Dzmura, and D. Anderson: A system that allows a physically handicapped person to use eye movement to efficiently control high quality speech output.

## VISION

A Braille Word Processing System, R. Stepp: A full powered braille word processing system whereby braille is keyed in, revised by a text editor and ultimately embossed by a computer-adapted Perkins brailler.

Kurzweil Reading Machine for the Blind, R. C. Kurzweil: Converts ordinary printed or typed material into full-word synthetic speech, enabling blind and dyslexic persons to read books, correspondence and other materials.

Keep in mind that these are developments that are already complete. The shame lies however, in that many were designed by private individuals or research groups and, because they are not "cost effective" may never reach those that could benefit most from them.

That the microcomputer can serve the handicapped by offering increased independence in communication, environmental control, recreation and community awareness, I don't think can be disputed. We are though in great need of "centres" skilled in the area of evaluating the handicapped in terms of the technical aid most beneficial to the individual and the most efficient means by which he/she could access it. Providing such evaluation centres would ensure maximum independence by the individual long before he reaches the classroom.

Adapted microcomputer availability in the schools, use of Teledon as an integral part of the educational curriculum for your "home or instituition" bound student and the education of group-home workers and educators in the use of this equipment are just some of the concerns that need to be addressed in the near future.

## ABSTRACT

The flexibility inherent in the microcomputer has resulted in a technical aid that, through software developments and/or hardware alternatives to the standard keyboard, can offer the handicapped greater independence in areas of communication, environmental control, recreation and education. Although some of these adaptations are available and in use in our classroom, progress has been as a result of a very fragmented, trial and error approach. A glimpse of the microcomputers' future potential in the field of technical aids was evidenced at the John Hopkins contest on "Microcomputers and the Handicapped". Hardware adaptation to allow the handicapped community access to Teledon, the availability of technical aids within the school, the training of group-home workers and educators in the use of adapted microcomputers, and access to professional "technical aid evaluation and training centres" represent just some of our current concerns.

# CREATING CAI FOR CLINICAL LABORATORY

Tracie M. Jenkins

Medical Center of Central Georgia

## ABSTRACT

Utilizing the Medlab CAI language PLAIN (Programming Language for Authoring Instructional Notation) and the CDC Hardware (System 17) also from Medlab, a division of CDC, personnel at the Medical Center of Central Georgia (MCCG), without benefit of a sophisticated team effort, created interactive CAI modules. They were inculcated into the laboratory's in-service education program and the medical technology student practicum. The modules were either self-contained or, for added versatility as required in morphology study and in learning how to use a sophisticated instrument, interphased with a laboratory instrument or a Singer 3300 Caramate. Early computer instruction generally consisted of a lesson, maximum length 25 minutes, and a quiz followed by a computer managed evaluation of the module. Later a pre-test as well as a post-test was included which, when used in conjunction with the evaluation, gave more insight into user learning and lesson effectiveness.

CAI was an extra undertaking, that took an individual weeks or months to complete. Ten tutorial lessons were created from September, 1980 to December, 1982. They were used and generally accepted as an integral part of the laboratory's educational program. How CAI was inculcated into the clinical laboratory and what was learned by the developers in terms of text and graphic design, interphasing and evaluation was the subject of this paper.

## INTRODUCTION

The Medical Director of the Medical Technology Program, Dr. Charles Maurizi, in September, 1980, initiated interactive computer instruction for the laboratory at the Medical Center of Central Georgia with the completion of the tutorial lesson entitled "Basic Coagulation". Although this lesson lacked the interphasing and graphic sophistication experience dictated designed into future lessons, it opened the door to the development of CAI for our laboratory.

It has been established that CAI tutorials are effective educational tools (Sprecker and Chambers, 1980). They are designed for the television generation of "visual learners" who respond to auditory cues to look and learn (Unknown, 1981). It was therefore the goal of Dr. Maurizi and those that worked with him to develop and incorporate this educational tool into the laboratory so that it would be utilized and, through use, win acceptance (Jenkins, Danker, 1981; Mitzel, 1981).

We realized that software development would not be easy or fast since CAI would be an extra project tacked on to our workload. We also wanted to create quality which meant progress would be slow, very slow. We had to minimally acquire skill in utilizing the CAI language PLAIN, develop content objectives leading to lesson design and visual layout, learn to program graphics into a system not designed for graphics, and design a computer managed evaluation. As Bonnie Seiler (1981) pointed out, "it is very rare that one person is highly competent in all these areas" and we were no exception. But neither were we confined by preconceived ideas about what we could not do. We were determined to develop well designed, interactive tutorial computer instruction that would be utilized by student and staff.

As we developed we made mistakes but we learned. We learned not to be verbose but succinct in our textual display. We learned from hours of playing with the

CAI language that periods could create faces, asterisks could represent tubular absorption, and dashes could create slides. We learned the rewards of interphasing slides and machines with the computer to enhance instruction. We learned from peer review and user feedback that lessons over 25 minutes were too long. We learned that creating tutorial CAI was difficult, very time consuming, and effective.

## TEXT AND GRAPHIC DESIGN

Incorporated into the PLAIN computer-based authoring system, part of the Tutor Computer-Assisted Instructional package designed to deliver PATHLAB concepts, was the basic instructional format found in other systems. It consisted of text presentation, questions, and feedback as well as branching possibilities. Once preliminary planning regarding objectives, content, expectations, and time, were completed lesson design was begun often utilizing the computer as the word processor. Initially authors filled the screen with text. As developers became more familiar with the programming language, verbage decreased in favor of a more inviting visual display. Manipulating the CRT screen to the best visual advantage without sacrificing the established order of the presentation, which our CAI users came to expect, naturally extended to graphics. It was clear that graphics could be used to complement text material, add reinforcement, and create expectancy as well as stimulate the user to recall prior learning (Gagne, 1981). Learning to create effective looking graphics was difficult on our system since it was not designed for graphics. As difficult as it was graphics proved to be an effective partner to text display in guiding learning, eliciting performance from the user and providing informative feedback which, hopefully, enhanced material retention and learning transfer (Gagne, 1981).

The following six general rules of design resulted from our experiences:

1. Text material should be arranged so that it is not only visually pleasing but had reasonable margins to facilitate comfortable reading.
2. Text per frame should be kept at a minimum.
3. Text and questions and/or answers in the same frame should be separated by a clear space.
4. Graphics should be clear and must complement the text.
5. Graphics such as a word written in all capitals or with spaces in between letters should be used sparingly.
6. Do not use flashing words, letters, numbers, or other such graphics unless it will enhance the lesson since this can create distraction.

These proved to be easy to understand, utilize and inculcate.

## INTEGRATING CAI INTO THE LIFE OF THE LABORATORY

CAI had to be integrated into the life of the laboratory if we expected it to be used and in return accepted (Jenkins and Dankert, 1981). We decided, therefore, to make it an integral part of the Medical Technology Practicum Curriculum that we were developing as well as a required part of our in-service education program.

Initial lesson topics were chosen that would fill a need both for students and personnel. Medical technology students came to the practicum experience week in Hematology, particularly in morphology. Also, technologist working in other areas such as Microbiology or Chemistry, and new personnel, needed a non-threatening form of review. This information plus an added established didactic need for student programs in renal and thyroid function provided us with numerous possibilities which were translated into lessons (Table 1).

Once the module (lesson and quiz) was completed, personnel were requested to take the lesson. Initially there was a lot of suspicion and related grumbling regarding CAI. As people used the system, however, they came to appreciate its non-threatening, self-paced aspect and even began to like it (Table 4). CAI lessons were used to compliment in-service presentations. They were also used for continuing education, which was required in our laboratory. Facilitating the needs of the employee was added incentive for them to go through the lessons.

CAI was integrated into the didactic, clinical and testing areas of the Medical Technology Curriculum. Just as students were specifically required to read a procedure, perform a clinical test or to take a written exam, so they were required to use CAI. For example, students were required, as part of their Special Chemistry rotation, to go through the Basic Renal Function and Disease Module (Table 1). The post-test counted as one of their exam grades. Such innovations proved to be very effective in getting students to use and trust the system. It also allowed the teaching technologist to turn some of his/her repetitive teaching responsibilities over to the ever-patient computer.

TABLE 1

CAI Developed as of February, 1982, listed in Chronological Order

| Lesson | Average Time | Pre-Test | Post-Test | Graphics | Interphased Material |
|---|---|---|---|---|---|
| Basic Coagulation #1 | 20 min. | | X | | |
| Basic Coagulation #2 | 15 min. | | X | | |
| Introduction to Blood Collecting | 10 min. | | X | | |
| Basic Thyroid Function | 25 min. | | X | | |
| Granulocytes | 20 min. | | X | | Caramate |
| Leukemia, One Classification | 25 min. | | X | | Caramate |
| Evaluation of the Peripheral Smear | 15 min. | | X | | Caramate |
| Basic Renal Function and Disease | 25 min. | X | X | X | Caramate |
| Use of the Hematrak | 25 min. | | X | X | Hematrak |
| Gram Stain, Process and Application | 20 min. | X | X | X | Caramate |

TABLE 2

Computer Managed Evaluation (CMV) Results[a] for the Module[b] Leukemia, One Classification

| No. | Evaluation Questions | Total Responses | Percent Response Rounded to Nearest Whole Number | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Does Not Apply | I disagree Strongly | I Disagree | I am Undecided | I agree | I agree Strongly |
| 1. | The subject matter covered in this lesson was too difficult | 14 | | 14 | 79 | | 7 | |
| 2. | The material covered in the quiz(s) was also covered in the lesson | 14 | | | | 7 | 93 | |
| 3. | The lesson material was confusing and hard to follow | 14 | | | 86 | | 14 | |
| 4. | Too much material was presented in this lesson | 14 | | | 71 | | 22 | 7 |
| 5. | It took too much time to complete this module (Lesson and Quiz). | 14 | | | 71 | 7 | 22 | |
| 6. | The quality of the slides shown with this module were very good. | 14 | | | | | 93 | 7 |
| 7. | The quality of the Sound Track Accompanying this Module was very good. | 14 | 93 | | | | 7 | |
| 8. | It was not difficult to find the time to complete this Module (Lesson and Quiz.) | 14 | | | 57 | | 43 | |
| 9. | This was a very effective way to learn the material presented | 14 | | | | | 100 | |
| 10. | I liked using the computer to learn or review educational material | | | | | | 93 | 7 |

[a] As of February 22, 1982

[b] Consist of a Lesson and a Post-test

## EVALUATION

We were grateful, not being degreed educators schooled in the theory and practice of educational technique, for the Medlab Computer-Assisted Instruction Development Package that provided for computer managed evaluation (CMV). At the end of each module, which was after the lesson's post-test, a request for the user to complete the evaluation was made. If the user agreed by answering yes, he/she was branched to the evaluation. Questions, designed to determine user attitude toward the module (Tables 2, 3, 4) had been standardized at the beginning of CAI development and thus, became static. Evaluation questions were designed to cover present requirements and future needs. If, therefore, a question was not pertinent to a module the user marked "Does not Apply". A histogram was printed of the responses received for each evaluation question. It indicated the percentage level of each response, the mean, and the total number of responses received. Also available were accumulated quiz scores, registered to date, with computer mean and number of uses for each, accumulated total time spent in each lesson as well as mean time and number of uses for each lesson.

CMV results provided us with valuable information. We quickly learned, however, that the results had to be interpreted carefully. If the user carriage returned without chosing one of the six responses the "I am undecided" response was automatically credited. It is difficult to know if the user meant to answer questions two and five in Table 2 as "I am undecided", since it represents a single response, or if he/she accidentally hit the carriage return. Also, if the user either misreads the question or presses the wrong key by mistake, an incorrect response is registered. Since we had not interphased sound, the reaction "Does not Apply" to question seven in Tables 2 and 3 is expected. The other responses to that question are not.

As we became more sophisticated CAI developers we realized that we could not determine student or user learning just with a post-test, which gave only pass or fail information. A pre-test as well as a post-test was included in later modules. This information added to our total evaluation process by enabling us to better determine the effectiveness of the lesson (Table 5). It was decided, therefore, that pre-test should be considered for modules that had been developed earlier.

Evaluation was a very important part of the CAI development program whether it was formal, characterized by Tables 2, 3, 4 and 5 or informal, characterized simply by asking our users what they liked or did not like. The lessons had been developed for our personnel and our medical technology students. They would use them if they perceived they were of value and they would not if they did not. When verbal feedback indicated users liked auditory cues and explanations to answers, we responded in kind.

Generally, as Table 4 demonstrates, our users thought that the subject matter covered in the lessons was not too difficult, the amount of material presented was within reason and it did not take an unreasonable amount of time to complete a module. Although many found it difficult to find time to complete the modules, most considered CAI as an effective way to learn and liked using the system.

## CONCLUSIONS

Realizing the computer's educational potential for the laboratory at the Medical Center of Central Georgia, Dr. C. Maurizi spearheaded a drive for CAI development. Primarily an individual endeavor requiring added commitment, it took weeks or months to complete one lesson. It was learned that what Seiler (1981) pointed out for CBI also applied to CAI.

> The development of effective computer-based educational materials requires expertise in many areas, including knowledge of the content; familiarity with student needs, interests, and abilities; ability to apply to CBE the principles of educational psychology dealing with effective instruction and evaluaion; familiarity with the capabilities and limitations of the CBE system; knowledge of good programming techniques, documentation, and the computer language, ability to construct graphics within the limitations of the computer medium; and ability to communicate in writing.

Knowledgeable laboratory personnel, utilizing the basic tutorial structure inherent in the MEDLAB CAI software, spent the weeks and months necessary to develop computer instruction (Table 1). Lessons, submitted to informal peer review, were edited, rewritten and finally released for laboratory staff and Medical Technology students to use. Formal computer managed evaluation coupled with informal evaluation added to the developers understanding of the needs, likes and dislikes of the users which caused lessons to improve. Simple rules for text and graphic design evolved, CAI was utilized and ac-

TABLE 3

Computer Managed Evaluation (CMV) Results[a] for the Module[b] Basic Renal Function and Disease

| No. | Evaluation Questions | Total Responses | Percent Response Rounded to Nearest Whole Number | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Does Not Apply | I disagree Strongly | I Disagree | I am Undecided | I agree | I agree Strongly |
| 1. | The subject matter covered in this lesson was too difficult | 6 | | | 83 | 17 | | |
| 2. | The material covered in the quiz(s) was also covered in the lesson | 6 | | | 17 | | 83 | |
| 3. | The lesson material was confusing and hard to follow | 6 | | | 100 | | | |
| 4. | Too much material was presented in this lesson | 6 | | | 50 | 17 | 33 | |
| 5. | It took too much time to complete this module (Lesson and Quiz). | 6 | | | 50 | 33 | 17 | |
| 6. | The quality of the slides shown with this module were very good | 6 | | | 17 | | 83 | |
| 7. | The quality of the sound track accompanying this module was very good. | 6 | 83 | | | 17 | | |
| 8. | It was not difficult to find the time to complete this module (Lesson and Quiz) | 6 | | | 50 | | 33 | 17 |
| 9. | This was a very effective way to learn the material presented | 6 | | | | 17 | 83 | |
| 10. | I liked using the computer to learn or review educational material | 6 | | | | | 83 | 17 |

[a] As of February 22, 1982

[b] Consisted of a Pre-test, Lesson, Post-test

TABLE 4

Computer Managed Evaluation (CMV) Composite of All Modules Results for Selected Questions[a]

| Evaluation Questions | Total Responses | Percent Response Rounded to Nearest Whole Number | | | | | |
|---|---|---|---|---|---|---|---|
| | | Does Not Apply | I disagree Strongly | I Disagree | I am Undecided | I agree | I agree Strongly |
| The subject matter covered in this lesson was too difficult | 79 | 6 | 13 | 65 | 13 | 3 | |
| Too much material was presented in this lesson | 78 | 8 | 9 | 63 | 13 | 6 | 1 |
| It took too much time to complete this module (Lesson and/or Quiz(s) | 78 | 1 | 6 | 67 | 18 | 8 | |
| It was too difficult to find the time to complete this module (Lesson and Quiz) | 78 | 3 | 4 | 27 | 17 | 46 | 3 |
| This was a very effective way to learn the material presented | 78 | 3 | 1 | 1 | 18 | 71 | 6 |
| I liked using the computer to learn or review educational material | 78 | | 1 | | 12 | 77 | 10 |

[a] As of February 22, 1982

TABLE 5

Review of Medical Technology Student Performance on the Pre and Post-Tests for the Lesson Basic Renal Function and Disease[a]

| Students Identified Numerically | Pre-Test Percent Score | Post-Test Percent Score |
|---|---|---|
| 1 | 40 | 80 |
| 2 | 60 | 80 |
| 3 | 60 | 90 |
| 4 | 30 | 80 |
| 5 | 20 | 90 |
| 6 | 60 | 90 |
| 7 | 60 | 90 |
| 8 | 50 | 90 |

[a] Represents students who did the module through February, 1982

cepted the more it was integrated into the life of the laboratory and computer managed evaluation continued to give added insight into lesson effectiveness (Table 4) as well as user attitude (Tables 2, 3, 5).

CAI development continued to be an ancillary responsibility and, therefore, work progressed at a very slow pace. The obvious alternative to this production dilemma was to create a network among MEDLAB users with CAI software. This was initiated by Dr. Maurizi under the acronym HOMER (Hospital Oriented Medical Education Resource). Archbishop Bergan, Mercy Hospital in Omaha, Nebraska was the first MEDLAB user to receive our lessons. They were enthusiastic about utilizing the lessons and modules in their Medical Technology Program as well as eventually developing some lessons of their own.

It has been demonstrated that computer instruction can effectively be created to serve the needs of a clinical laboratory. Also described was how efforts to increase CAI offerings resulted in two hospital laboratories sharing computer instruction. They gave birth to a promising child - HOMER.

## REFERENCES

Gagne, R.M., Wauger, W., & Rojas. Planning and authoring computer-assisted instruction lessions. Educational Technology, 1981, 21(9), 17-26.

Jenkins, T.M. & Dankert, E.J. Results of a three-month PLATO trial in terms of utilization and student attitudes. Educational Technology, 1981, 21(3), 44-47.

Mitzel, H.E. On the importance of theory in applying technology to education. Journal of Computer-Based Instruction, 1981, 7(4), 93-98.

Seiler, B.A. Applying a systematic approach to CBE development in an academic environment. 1981 Conference Proceedings of the Association for the Development of Computer-Based Instructional Systems, 1981, 305-310.

Sprecher, J.W. & Chambers, J.A. Computer assisted instruction: factors affecting courseware development. Journal of Computer-Based Instruction, 1980, 7(2), 47-57.

Unknown. Into the electronic learning era: implications for education and psychological research. An interview with Mary Alice White. Educational Technology, 1981, 21(9), 9-13

# TOWARD NATURAL LANGUAGE AUTHORING OF LESSONS ON PLATO

K.M. Bagnall, M. Szabo, I. Rollingher, and

A. Kallal, The University of Alberta

## ABSTRACT

This paper addresses the issue of using the resources of a mainframe CBL system, subject matter specialist and analyst skills to produce software aimed at reducing time involved in creating CBL materials without reducing quality.

Our previous studies have shown that first year medical students of Anatomy received well a small series of clinical-simulation lessons in which the relevance of Anatomy was highlighted in practical situations. The lessons incorporated many standard PLATO features but the rate of production for these lessons was painfully slow. It was recognised that each of the clinical simulations could be made to follow a similar sequence and accordingly an authoring lesson was developed. The concept of variability was stressed throughout the development of the authoring lesson so that the resulting lessons had variety and remained attractive whilst still retaining the anatomical and medical information.

## INTRODUCTION

It has been reported that an observer of one of the early automobile mass production facilities pondered and asked "where will we ever get enough chauffeurs to drive all these cars?". This statement is perceptive as it conveys an engineer's entry into the world of social and political change driven by technology. With the corresponding explosion of powerful computers in recent years, one may ponder a similar problem, "Where will we ever get enough systems analysts and programmers to drive these things?" The current solution of hiring from another employer has obvious upper limits which we may be rapidly approaching.

In the area of producing computer based learning (CBL) materials, the problem is intense due to; 1) costs; 2) unavailability; and 3) lack of training of programmers. This paper addresses the issue of using the resources of a mainframe CBL System subject matter specialist and analyst skills to produce software aimed at reducing time involved in creating CBL materials without reducing quality. As Thornburg stated, "Designers must now think about saving time for people, not for the computer" (Heller, 1981). Experience with a PLATO lesson designed for use by a subject matter specialist with no CBL experience is highlighted in this paper.

## PROBLEM

In a study by Mah (1980), a small series of clinical-simulation lessons were created for use on the University of Alberta PLATO computer in which the relevance of the Anatomy taught to first year medical students in their basic science studies was highlighted in practical situations. These lessons incorporated many standard PLATO features which resulted in the inclusion of drawings, patient-doctor dialogue, touch panel and questions and answers, and were well received by the students. The lessons were an attempt to overcome the attitude of Anatomy being a morbid subject and instill the idea of Living Anatomy with an emphasis on its importance to a practising clinician.

A recent study by Rollingher (1981) attempted to increase the number of clinical-simulation lessons. Several items quickly became obvious.

1. The rate of production was slow. There are many clinical situations which could be replicated using PLATO but their translation into lessons re-

quires much more than a superficial knowledge of TUTOR language. A barrier exists between those people who have knowledge to provide the educational material and the production of actual lessons.

2. Within limits, each of the clinical simulations could be made to follow a similar sequence; a) introduction - setting the scene; b) a patient with a problem; c) a question and answer session based on the problem; d) a diagnosis.

A search for similar problems revealed that several major companies, such as Control Data and Texas Instruments have recognized the need to reduce escalating costs associated with expensive and scarce subject matter specialists, programmers, and instructional designers. They have laid plans to produce natural language intessigent CBL software which permits the development of high quality courseware with minimum source, designer, and programmer time.

Although high level CBL software has been proposed in the past, the availability of well developed software authoring languages has made such a development a difficult task. Another phenomenon driving the powerful interest is declining hardware costs which have attracted many 'educational moths' to the flame of CBL. Escalating courseware and software development costs have applied heat and many 'moths' have gone down in flames.

## ONE SOLUTION

From these observations emerged the idea of producing an "author lesson" for the creation of other lessons. A subject matter specialist wishing to present educational material in the form of PLATO lessons could do so by following simple instructions given by the author-lesson. The created lessons would be developed on a display-by-display basis. The following page options are now available:

1. A title page. A standard title display with space for identification of the lesson.
2. A display of text a) as 3 paragraphs b) as 2 paragraphs c) as 1 paragraph. Using NEXT to deliver paragraphs.
3. A text display plus instructional graphics - primarily to introduce the patient pictorially including his symptoms.
4. A question display plus picture. The screen is completely touch active and the student is asked to identify structures by touching it on the screen. The option to enter feedback is also included.
5. A question display plus touchboxes. A student responds to a question by touching a box associated with an appropriate response. Appropriate feedback can be given for every box. The position and number of boxes can be designed by the author.
6. A synopsis display. This is a brief summary of the clinical situation which is available to the student at any time by pressing a particular key.
7. A review display. This reviews the main features of the lesson and is available at any time by pressing a particular key.

The author-lesson introduces the displays in template form and the lesson creator follows simple instructions for the creation of each display. The available displays involve as much flexibility as possible and can be added in any order an infinite number of times. The author-lesson also allows a lesson to be created over several separate editing occasions and also allows the creator to step-edit through the lesson allowing for the insertion or removal of individual displays.

For the displays which include pictures a graphics display library was created from which the lesson creator could borrow relevant graphics following simple instructions from the author-lesson. Once called from the library, the graphics can be moved around the display and altered in size within a window, allowing only parts of pictures to be drawn if required.

Recent trials of the author-lesson with people who have no knowledge of TUTOR language have been highly encouraging with lessons being created very easily. Attention has been moved from the difficult programming side to the educational aspects of the lesson. The most important result of this project is the elimination of the inhibiting barrier relating to the creation of PLATO lessons, with the only requirement now being a basic understanding of PLATO lesson construction.

The author-lesson will not be restricted to any one discipline. Its flexibility makes it appealing to a wide range of potential authors and if any discipline required "special" displays, these could be created and added to the available options. We, ourselves, are continuing to develop more option-pages, particularly in the "teaching" area.

One important feature of this program is that the design is being led by the educators in close association with the programmers. The educators have an understanding of PLATO and this eliminates the development of unnecessary material which could inhibit the development of lessons.

Several empirical questions remain to be answered and it is our intention to apply ourselves to their solution. Foremost, do the anatomy lessons generated with this system cause meaningful learning to occur? Second, what is the extent of time (design, development, production and validation) actually saved? Finally, what problems are encountered with use of this system in areas beyond anatomy? Our primary efforts have focussed on the construction of the software program; answers to these questions must remain for the future.

## CONCLUSIONS

This continuing project began when one faculty member devoted sufficient time to creating PLATO lessons to see beyond to the greater problem of barriers to extensive involvement by faculty. If the 'avoidance reaction' of subject matter specialists producing CBL materials could be reduced by natural language authoring systems, the quantity and quality of lessons could be expected to improve.

Tradeoffs are, of course, expected. As faculty authors become more interested and knowledgeable, their level of instructional sophistication will rise along with their demands on the system. All we can hope for is to continue to increase the ability of the system to handle increasingly sophisticated demands commensurate with the growth of authors. In the meantime, we hope to deal with the wide spread 'avoidance reaction' characteristic of much of education and training today.

One may ask in anticipation of the expected developmental demands, "where should we be heading?" The following broad system specifications will be examined closely in the future.

### Natural Language

We have done a credible job in producing turnkey lessons which use natural language and interact with the student in a tutorial fashion. However, most applications have been of the computer 'programming' the student, rather than the student teaching the computer, to paraphrase Papert (1980). We currently need turnkey systems for instructors which enable them to teach the computer to teach.

### Query Based

The system should query the teacher and guide him/her in the production of lessons. Two kinds of questions are essential for the author: "What do you want to do next?" and "Do you want some help in doing it?" These apparently open-ended questions can be modified for an effective query based author-lesson.

### Text and Logic Editors

The two essential components of CBL materials are text (used in the broadest sense of the word), and branching logic. These two components should be capable of easy interactive editing to enable the instructor to share a close interaction with the lessons being authored.

### Instructional Strategies

The current author-lesson described above is designed to produce lessons of a specific but flexible instructional strategy. An author-lesson to permit utilization of a wide range of strategies is an admirable goal and new developments to approach that goal will be watched carefully. At the University of Alberta, a second group has stepped forward to propose a similar author-based lesson to produce a specific type of patient management simulation patterned after the work of McGuire et al. at Illinois.

### Subject Matter Independence

We do not have a corner on the specifications which make an author lesson transportable across subject matters. Certainly the ability to specify unique text and graphic libraries is crucial. Perhaps several successful demonstrations will be sufficient for transportability.

### Interactivity

The ability to create small units or modules and run them immediately to produce author and student reactions is more desirable than the lengthy time that now exists between lesson development and student use. One wonders whether the time delay of days or weeks negatively impacts the objectives of the lesson, the author's motivation to continue, or both. Lessons should be runable within hours of creation.

### Mainframe Size

Our initial exploratory efforts at producing author-lessons have required the power of a large mainframe system. Certainly, the automatic compiling of code from natural language input and generation of lesson drivers needs powerful language capabilities and memory storage available today in large mainframe systems. We expect that as our knowledge and abilities to produce code increase, as software systems increase in sophistication from artificial intelligence/natural language processing research, and hardware capacity increases relative to size and cost, the picture may change.

At the present, the turtle 'program' from Project LOGO is now available for implementation on small systems. The trend seems to be clear. So too, the ultimate author-lesson, running on a large mainframe will produce lessons, which when validated, will run downline-loaded onto small stand-alone units.

## REFERENCES

Heller, D.K. User-friendly languages of the future. Interface Age, 1981, 78-82.

Mah, L., How can PLATO be used as an adjunct to teaching Anatomy? Summer Research Project, University of Alberta, 1980.

Papert, S. Mindstorms: Children, Computers, and Powerful Ideas. New York: Basic Books, 1980.

Rollingher, I. PLATO in Anatomy. Summer Research Project, University of Alberta, 1981.

# THE CAI MICROCOMPUTER PROJECT IN THE FACULTY OF NURSING AT THE UNIVERSITY OF CALGARY

K. J. Hannah, D. N. Conklin

The University of Calgary

In 1976, the University of Calgary Faculty and Nursing decided to investigate the potential of Computer Assisted Instruction (CAI) as an alternative teaching strategy for baccalaureate nursing students. The impetus for this decision came primarily from the faculty's "Statements on Learning and the Learner" which are found in its Conceptual Framework. This document states that "Each learner has individual learning needs. These needs necessitate a variety of creative teaching strategies to achieve motivation toward potential".[1]

Added impetus came from recognition of the fact that there are vast numbers of nurses who are seeking baccalaureate education in nursing but who live in geographical areas which are remote from the three traditional Alberta universities which offer baccalaureate degrees in Nursing. In order to meet the learning needs of these students, alternative teaching-learning strategies needed to be identified in order to facilitate the most effective, efficient and appropriate use of faculty and student resources while maintaining the quality of the nursing programme at The University of Calgary.

A four phase plan was developed. Phase I was to investigate the effectiveness of CAI as an alternative teaching strategy in our setting. The objectives in this phase were:

1) To evaluate the learning which occurred, the retention of learning and the transfer of learning to the clinical area;

2) To evaluate the resources available from The University of Calgary Computer Serivces Division; and

3) To evaluate the student's receptivity to this mode of learning.

Phase II of the plan included:

1) Identification of areas in the curriculum which could be effectively taught using CAI;

2) Development of a catalogue of available CAI lessons in nursing, and

3) Integration of presently available CAI lessons into appropriate locations in the curriculum.

Phase III of the plan, the goals were:

1) Addition of computer terminals to the Learning Resource Center;

2) Acquisition, purchase, and translation of additional CAI programmes from sources identified in Phase II;

3) Adaption of programmes identified in Phase II to meet the objectives of The University of Calgary nursing curriculum;

4) Development of new CAI programmes in nursing at The University of Calgary.

The fourth, and final phase of the plan consisted of developing and making available CAI-based self study courses for nurses in rural Alberta by which credits towards the Bachelor of Nursing degree at The University of Calgary could be acquired.

A time line was not developed because the phases were conceptualized as being concurrent rather than sequential.

## Evaluation

Progress of the four phase plan was evaluated in the Spring of 1981 and is summarized in Figure 1.

### Phase I

Both a pilot study and a major research study have been conducted and the results published elsewhere.[2,3,4] These studies produced significant positive findings with regard to the learning which occurred and the students' attitudes to this type of learning. However, student frustration resulting from the limited number of available terminals, system failures, and slow response times was identified. In 1978 the campus mainframe was changed from a CDC Cyber 172 to a Honeywell Multics with the result that 18 months were spent converting and debugging lessons. In spite of this change, turn around and down time with the new system were unchanged and no additional terminals were available. Still another major system "upgrade" is currently being negotiated. We also encountered major difficulties contracting for programme services through the campus Computer Services whose technicians/programmers were unfamiliar with the nursing terminology and concepts, and with the computing needs of our faculty.

### Phase II

The curriculum was reviewed in light of the Faculty's philosophy of self-directed, individualized learning and areas were identified which could be effectively taught using CAI learning modules. CAI lessons were designed, programmed, and integrated into several courses within the baccalaureate curriculum. Preliminary contacts were made with other nursing faculties using CAI and major sources of nursing CAI were identified. However, the notion of developing a catalogue of existing nursing CAI was deferred. This deferral was due to such obstacles as hardware and software incompatibilities, lack of sufficient human and financial resources to institute mechanisms to assure quality programmes, and the lack of sufficient documentation to permit evaluation of congruence between the CAI lesson objectives and curriculum objectives.

### Phase III

We succeeded in acquiring two (2) terminals for use in our Learning Resource Center and in completing a library of eleven CAI lessons available on the Multics system. During each of the past two academic years our instructional computing has multiplied by a factor of four. In the first 2/3 of the 1981-1982 academic year we completely exhausted all funds allocated to the Faculty of Nursing for the academic year by the University's Academic Computing Review Board.

These experiences convinced us that the concept of instructional computing was a viable alternative learning strategy, enthusiastically received by students and a working nucleus among the faculty. However, autonomy and control of hardware were essential to the success of widespread use of CAI within our programme. In light of recent technological developments and steadily reducing cost factors, we concluded that microcomputers were preferred for future CAI development in our situation.

### Current Happenings and Future Plans

Recently Monroe Systems for Business donated five (5) Monroe EC8800 Advanced Educational Microcomputers and peripheral equipment. The Advanced Education Endownment Fund of the Province of Alberta provided a matching grant which permitted us to purchase an amount of equipment equivalent to that which was donated. As a result the Faculty of Nursing at The University of Calgary now has within our Learning Resources Center a Microcomputer Learning Laboratory which contains:

10 EC8800 Monroe Colour Microcomputers
2 Accessory Disk Drives
2 Daisy Wheel Printers
2 Dot Matrix Printers
2 Voice Digitizers
2 Cluster Control Assemblies
2 Acoustic Couplers
Related Machine Software and Documentation

Our first task was to have the equipment installed in the new facilities to which the Faculty of Nursing recently moved. Technically, the hardware is very advanced and sophisticated, however, it is also very new to the education market. There exists, therefore, no CAI lesson software or authoring system. Further, no programs were available to interpret software designed to run on other vendors' microcomputers.

Conversion of our existing library of CAI lessons to run on this newly acquired hardware is being undertaken by a recently hired programmer/analyst. This person is responsible to the Director of the Learning Resources Center within the Faculty of Nursing. It is anticipated that this conversion will be completed in preparation for the 1982 fall University term. In the interim we are maintaining our lessons on the Multics System so that should the conversion process be prolonged, there will be no disruption of academic courses in which CAI had been previously integrated.

The long range plan for instructional computing within the Faculty of Nursing at The University of Calgary has been revised and updated as shown in Figure 2. There are three major components to this new five year plan:

1) Provision of support to faculty members who want to initiate or expand the use of CAI in their area of content expertise.

2) Integration of instructional computing at appropriate points in the curriculum.

3) Research and development.

### Support for Faculty

The main objective of this component is to assist interested faculty members towards greater computer literacy and awareness. This is being accomplished by providing faculty with regular opportunities to be introduced to and/or updated on the use of the hardware in the Microcomputer Learning Laboratory. We have also offered sessions for faculty on instructional design unique to CAI lesson development.

The second objective in component 1 is to provide technical support to faculty members who are developing CAI lessons. To this end we use the expertise of our recently highered programmer/analyst and have developed a close working relationship with the technical unit of the Office of Medical Education. This unit is responsible for coordinating CAI activities in the Faculty of Medicine using 46 microcomputers identical to ours. We anticipate that within the Learning Resource Center an instructional methodologies support team will be developed. As presently conceptualized an educational technologist, computer programmer/analyst, evaluation consultant, and medical technician will be available to assist faculty members (the content experts) in the development of multimedia instructional materials including CAI.

The third objective of this component is to fully document the local process for developing CAI materials. We envision using this documentation as a resource for faculty to guide them through the systematic approach to educational technology.

The fourth, but probably the most important objective is to acquire a user friendly, natural language authoring system. This is a joint objective of both the faculties of Nursing and Medicine and will also require vendor support and assistance. The primary system currently being considered is the CAI Authoring System developed by Dr. R. Pogue of the Medical College of Georgia in Pascal for the Apple II Microcomputer.

## Integration into the Curriculum

While waiting for the installation of the hardware, and during the conversion of our existing programme library, we proceeded to develop 11 patient simulations. These have all been tested and evaluated with nursing students as paper and pencil simulations. They are now ready for programming and we expect them to be available early in 1983 for use in our undergraduate nursing programme. At the same time we are committed to exploring with the Faculty of Medicine the modifications necesary to make these simulations useful in their teaching programme.

Another objective within the second component of the current 5 year plan is to assist faculty to enter their exam item banks onto diskettes in a system which will pernit random selection of items for generation of exams. This is not a new concept, but it will provide a previously unavailable curriculum resource for faculty members. We expect to also provide a system for storage of item analysis statistics and eventually perhaps to generate unique but fully equivalent forms of exams for each student. Ultimately this could form the nucleus for the development of a Computer Managed Instruction (CMI) system. The item bank for the course which is serving as a pilot project for this objective is currently being entered into the system.

On going review of the curriculum to identify areas which could be effectively taught using educational computing (Item 2.1 of the earlier plan) will continue. Priorities are being established in response to faculty requests.

## Research and Development

As mentioned earlier, ongoing local development of CAI materials for use in meeting course and curriculum objectives has high priority. Cooperation with the Faculty of Medicine in the development of a common data base of patient information for use in clinical simulations is accorded similar priority.

Acquisition or local development of a system for monitoring and evaluating student progress (CMI) is a more distant activity, but one which we do not intend to overlook.

Protocols for reviewing CAI lessons to assure quality, effectiveness, curriculum relevance (usefulness) and currency of content, are being established.

Obviously, the monitoring of ongoing technological developments is essential. Currently, a videodisk to interact with a CAI lesson which teaches the catheterization procedure is being developed locally. Our Learning Resource Center is following the development and faculty members have provided consulting services. Other research projects within our faculty are now being formulated to investigate learning styles and the use of microprocessors in health promotion among patients and their families.

## SUMMARY

It has been our intention to provide a brief glimpse of our experiences during the last five years, complete with successes and failures in the hope that we might save someone else some frustrations and hasten their successes. We recognize that we are neophytes and still have much to learn but we believe we are making progress and are sustaining our enthusiasm. Our ultimate goal remains the delivery to nurses in rural Alberta of CAI based courses for credit towards the baccalaureate degree in nursing at the University of Calgary.

FIGURE 1

Summary of Activities 1976-81

| | Initial Goal(s) | Related Activities | Evaluation |
|---|---|---|---|
| 1. | Demonstrate effectiveness of CAI as alternative teaching strategy in our setting. | Pilot project[2,3] and major research project[4] completed. | CAI effective; good faculty and student response; facilities inadequate. |
| 2.1 | Identification of areas in curriculum which could be effectively taught using CAI. | Surveyed on an ongoing basis. | See 3.3 below and Item 2.4 in Figure 2. |
| 2.2 | Inventory of Nursing related CAI existing in North America | Preliminary contacts made and major sources identified. | Deferred. |
| 3.1 | Addition of computer terminals to Faculty of Nursing, Learning Resource Center. | One C.R.T. and one thermal print hard copy terminal acquired. Faculty and student use have demonstrated need for more. | Microprocessors preferred. |
| 3.2 | Acquisition of CAI lessons from external sources. | Three lessons have been acquired and translated to run on the Honeywell Multics. Additional acquisitions await completed of 2.1. | Deferred. |
| 3.3 | Development of CAI lessons by The University of Calgary Faculty of Nursing. | 11 completed and available on Multics. | Eleven in progress. |
| 4. | Delivery to nurses in rural Alberta of CAI based courses for credit towards the baccalaureate degree in nursing at The University of Calgary. | Preliminary discussions re: delivery mechanisms undertaking. Demonstrated need for this purpose as opposed to telephone link to larger system. | Continue to negotiate. |

## FIGURE 2

## Plan for Instructional Computing 1982-1987

1. Faculty Support
   - 1.1 Computer literacy and awareness forums
   - 1.2 Instructional methodologies team
   - 1.3 Document local processes for authoring
   - 1.4 Authoring system
2. Curriculum Integration of CAI
   - 2.1 Finalize develop 11 new patient care simulations
   - 2.2 Item banking
   - 2.3 C.M.I.
   - 2.4 Identification of areas in the curriculum which could be effectively taught using CAI.
3. Research and Development
   - 3.1 C.M.I.
   - 3.2 Quality assurance protocols
   - 3.3 Monitor new technology
   - 3.4 Develop research projects.
4. Delivery of CAI based credit courses to nurses in rural Alberta

## Bibliography

1. "Statement on Learning and the Learner". Conceptual Framework. Faculty of Nursing, The University of Calgary, Calgary, Alberta.

2. Hannah, Kathryn J. "Report of the Computer Assisted Instruction Project in the Faculty of Nursing at The University of Calgary". New Directions in Educational Computing. J. Hirschbuhl (ed.), ADCIS Publications, Bellingham, Washington, 1978, pp. 289-294.

3. Hannah, Kathryn J. "Overview of Computer Assisted Learning in Nursing Education at The University of Calgary". Perspectives: Nursing Education, Practice, Research. C. Stainton e tal (eds.), The University of Calgary, Alberta, 1979, pp. 43-56.

4. Conklin, Dorothy. "A Study of Computer Assisted Instruction in Nursing Education". Unpublished Masters thesis, The University of Calgary, 1981.

# The Use of Computer Simulation to Teach Calculation and Regulation of Intravenous Flow Rates to Nursing Students

Donna E. Larson

Grand Valley State Colleges
Allendale, Michigan 49401

## ABSTRACT

A quasi-experimental study was conducted to determine if a computer simulation could effectively and efficiently teach the calculation and regulation of intravenous (IV) flow rates, a basic nursing skill with both psychomotor and cognitive components. The subjects were 48 basic baccalaureate nursing students enrolled in the Nursing II course at Grand Valley State Colleges in Allendale, Michigan, during Winter Semester, 1981.

Two research questions were addressed. First, was the computer simulation as effective as a laboratory simulation (the conventional teaching methodology) for instructing students? Second, was the computer simulation more efficient in terms of student learning-time and cost than the laboratory simulation?

All students were provided with initial instruction on calculating and regulating IV flow rates. The experimental group (N=24) practiced the skill using the computer simulation; the control group (N=24) practiced the skill using actual IV equipment in the skills laboratory simulation.

Students were tested on their abilities to calculate an IV flow rate and then to regulate an actual IV system so that it delivered the prescribed fluid volume. All students were tested both immediately following the initial treatment and ten weeks later. Data were analyzed with analysis of covariance and analysis of variance.

The conclusions based upon the findings were: (1) computer learners were able to calculate an IV flow rate and regulate an actual IV system as well as laboratory learners; (2) computer learners were able to complete significantly more practice problems in significantly less time than laboratory learners; (3) when costs of computer hardware and software were not included, and when instructional and research costs were combined, the cost per student for computer learners ($.94) was less than half the cost per student for laboratory learners ($2.17); and (4) when projected costs of computer hardware and software were amortized over three years, and when instructional costs were separated from research costs, the cost per student for computer learners ($1.44) remained less than the cost per student for laboratory learners ($2.01).

The computer simulation used in this study was developed by this investigator. The program was designed to be used on a TRS-80 Model I microcomputer and featured graphics and real-time animation.

## INTRODUCTION

The correct performance of basic nursing skills is an essential component of professional nursing practice. The mastery of such skills is very important to the safety and well-being of patients. The teaching of basic nursing skills is, therefore, a significant component of nursing education curricula.

One of these essential nursing skills is the calculation and regulation of IV flow rates. While performing this skill, the student must first use available patient and equipment data in carrying out three interrelated and sequential mathematical calculations. These calculations determine the flow rate of fluid to be intravenously administered to the patient. The student must then count the number of drops per minute infusing into the patient and adjust the equipment so that the system actually delivers fluid into the patient at the prescribed rate. The skill, therefore, has both cognitive and psychomotor components.

This skill is usually taught in a controlled laboratory setting, often called a "skills laboratory". The skills laboratory generally contains necessary actual equipment and supplies and serves as an environmental simulation of the clinical practice setting.

When teaching the calculation and regulation of IV flow rates in a skills laboratory, a faculty member usually presents the essential elements of the skill and then demonstrates the skill to a large group of students. Students then break into smaller groups to practice. The faculty member circulates among the groups, supervising individual performance, answering questions, and providing feedback. Students practice the skill during their assigned laboratory times and on their own, when the skills laboratory is not being used by other students or for other instruction.

Such laboratory instruction is usually cumbersome, time-consuming, and costly. Because faculty may be "spread too thin" to be aware of individual student needs, students may not receive needed feedback and additional instruction. Moreover, students themselves may not even be aware that they are performing a skill incorrectly, thus learning something that has to be "unlearned" later. If a student does require additional instruction and practice, it is often logistically difficult to provide.

This situation presents an opportunity for an alternative instructional strategy that would enable students to learn independently and reinforce their basic nursing skills, without the problems inherent in the usual skills laboratory. Such an alternative could have a significant impact on both the quality of nursing education and health care for the public.

Computer-assisted instruction (CAI) may provide such an alternative instructional strategy. Previous research in nursing education has demonstrated that CAI is effective and learning-time efficient for teaching cognitive and problem-solving skills (Bitzer & Bitzer, 1973; Bitzer, Boudreaux, & Avner, 1973; Droste-Bielak, 1980; Huckabay, Anderson, Holm, & Lee, 1979; Kirchhoff & Holzemer, 1979; Rubinson & Robinson, 1977-78; Valish & Boyd, 1975). However, this investigator has found no reports of research on the use of CAI for the teaching of basic skills which have a psychomotor component, either in nursing or in other health professional education.

The purpose of this study was to determine if a computer simulation developed by this investigator could effectively and efficiently teach the basic nursing skill of calculation and regulation of IV flow rates (a skill with both cognitive and psychomotor components). This purpose was delineated into two research questions. First, was the computer simulation as effective as the laboratory simulation? Specifically, were students who practiced the skill with the computer simulation able to calculate flow rates as correctly and regulate actual flowing IV systems as accurately as students who practiced the skill with the conventional laboratory simulation? Second, was the computer simulation more efficient than the conventional laboratory simulation? Specifically, did students spend less time learning the skill with the computer simulation than with the laboratory simulation?

This study also attempted to determine and compare the cost per student using the computer simulation with the cost per student using the conventional laboratory simulation. Costs were determined by purchases of consummable equipment and supplies and faculty time used in setting up, dismantling, and instructing in the IV calculation and regulation portions of regularly scheduled instruction, impromptu individual instruction, and "open practice sessions" of the laboratory and computer simulations.

## DESIGN AND METHODOLOGY

This study was a posttest-only, control group quasi-experiment, similar to Campbell and Stanley's (1963) static-group comparison design. The subjects were 48 basic baccalaureate nursing students enrolled in the Nursing II course at Grand Valley State Colleges in Allendale, Michigan, during Winter Semester, 1981.

Because of concerns about potential cross-contamination of subjects during the ten week period of the study, subjects were assigned to either the experimental or control group on the basis of their assignment to the regular skills laboratory. Students in the Monday skills laboratory were assigned to the experimental group (N=24), while students in the Tuesday skills laboratory were assigned to the control group (N=24).

This investigator presented classroom demonstrations of the calculation and regulation of IV flow rates to both treatment groups; these demonstrations followed a required student reading assignment on the topic. Students in the experimental group then practiced the skill for a maximum of 30 minutes using the computer simulation on TRS-80 Model I microcomputers. Students in the control group practiced the skill for a maximum of 30 minutes using actual equipment in the skills laboratory simulation.

After practicing with either the computer simulation or the laboratory simulation, all students were immediately tested on their ability to correctly calculate an IV flow rate and then to accurately regulate an actual IV system to deliver the prescribed volume of fluid. Ten weeks later, all students were again tested in the same manner.

Data were analyzed by analysis of variance and analysis of covariance. By using intact groups rather than random assignment of subjects to experimental and control groups, posttest differences between groups might be attributed to characteristics of the groups rather than to the treatment. It was believed that a student's prior mathematics ability and prior and concurrent clinical experiences in caring for patients receiving IV fluid therapy were the variables that would cause the greatest potential dissimilarity between treatment groups. Therefore, these variables were measured and treated as covariates to further equalize the groups.

## FINDINGS

Research Question One: Was the computer simulation as effective as the laboratory simulation for teaching students the calculation and regulation of IV flow rates?

On both the immediate and the delayed posttests, there were no significant differences between groups in their ability to accurately calculate IV flow rates (a cognitive skill). The computer learners were able to accurately calculate IV flow rates as well as the laboratory learners (see Tables 1 and 2). Data were analyzed by analysis of covariance. The score on a mathematics for medication dosage examination that all students took early in the semester was used as the covariate for the immediate posttest. The number of IV flow rate calculations performed clinically during the ten weeks between the immediate and the delayed posttests was used as the covariate for the delayed posttest.

Furthermore, on both the immediate and the delayed posttests, there were no significant differences between groups in their ability to accurately regulate IV flow rates (a psychomotor skill). The computer learners were able to accurately regulate an actual IV system as well as the laboratory learners (see Tables 3 and 4). Again, data were analyzed by analysis of covariance; previous experience in caring for patients receiving IV fluid therapy was used as the covariate for the immediate posttest. Clinical experience with regulating IV flow rates during the ten week interim between the immediate and delayed posttests was used as the covariate for the delayed posttest.

Table 1 -- ANCOVA for immediate posttest calculation score*.

| Treatment Group | Mean Calculation Score | Standard Deviation | F | Significance of F |
|---|---|---|---|---|
| Computer Learners | 2.792 | 0.588 | 0.420 | 0.52 |
| Laboratory Learners | 2.625 | 0.875 | | |

* Possible range of scores: 0 to 3

Table 2 -- ANCOVA for delayed posttest calculation score*.

| Treatment Group | Mean Calculation Score | Standard Deviation | F | Significance of F |
|---|---|---|---|---|
| Computer Learners | 0.833 | 0.381 | 1.040 | 0.313 |
| Laboratory Learners | 0.708 | 0.464 | | |

* Possible range of scores: 0 to 1

Table 3 -- ANCOVA for immediate posttest regulation score*.

| Treatment Group | Mean Regulation Score | Standard Deviation | F | Significance of F |
|---|---|---|---|---|
| Computer Learners | 2.917 | 0.282 | 0.345 | 0.560 |
| Laboratory Learners | 2.833 | 0.637 | | |

* Possible range of scores: 0 to 3

Table 4 -- ANCOVA for delayed posttest regulation score*.

| Treatment Group | Mean Regulation Score | Standard Deviation | F | Significance of F |
|---|---|---|---|---|
| Computer Learners | 2.875 | 0.448 | 0.158 | 0.693 |
| Laboratory Learners | 2.917 | 0.282 | | |

* Possible range of scores: 0 to 3

Research Question Two: Was the computer simulation more efficient in terms of student learning-time and cost than the laboratory simulation?

Throughout this study, records were kept of both the amount of time students spent practicing calculating and regulating IV flow rates and the number of practice problems/scenarios they worked through. Data were analyzed by analysis of variance.

The computer learners spent significantly less (at the 0.05 level) time practicing calculating and regulating IV flow rates than did the laboratory learners (see Table 5). Furthermore, the computer learners worked through significantly more (at the 0.05 level) practice problems than did the laboratory learners (see Table 6). In other words, the computer learners practiced significantly more in significantly less time than the laboratory learners; the computer simulation was more efficient in terms of student learning-time.

Table 5 -- ANOVA for time spent practicing the skill.

| Treatment Group | Mean Minutes Spent | Standard Deviation | F | Significance of F |
|---|---|---|---|---|
| Computer Learners | 23.6 | 6.14 | 4.35** | 0.0425 |
| Laboratory Learners | 28.8 | 10.3 | | |

** Significant at 0.05 level.

Table 6 -- ANOVA for learning scenarios students worked through.

| Treatment Group | Mean Number of Learning Scenarios | Standard Deviation | F | Significance of F |
|---|---|---|---|---|
| Computer Learners | 3.71 | 1.73 | 25.75** | 0.0000 |
| Laboratory Learners | 1.58 | 1.10 | | |

** Significant at 0.05 level.

The computer simulation was also more efficient in terms of cost per student. As can be seen from Table 7, the cost per computer learner was $.94, while the cost per laboratory learner was $2.17. It should be noted, however, that these costs reflect a combination of costs incurred for instruction and the special requirements of this research study. Also, these costs do not reflect the costs of computer hardware and software, since the research setting already possessed the necessary microcomputers and program. When instruction and research costs are separated, and when projected costs for computer hardware and software are amortized over three years, the cost per computer learner is $1.44, while the cost per laboratory learner is $2.01 (see Table 8). Therefore, even when computer equipment costs are included, the computer simulation remains less costly than the laboratory simulation.

Table 7 -- Costs of faculty time and consummable materials (combined instructional and research costs).

| Treatment Group | Faculty Time ($18000/yr av. salary) | Intravenous Equipment | Total Cost | Cost per Student |
|---|---|---|---|---|
| Computer Learners (N=24) | $17.28<br>(120 min) | $5.21<br>(1 set) | $22.49 | $ .94 |
| Laboratory Learners (N=24) | $31.25<br>(217 min) | $20.84<br>(4 sets) | $52.09 | $2.17 |

Table 8 -- Comparison of projected costs for instruction alone.

| Treatment Group | Faculty Time ($18000/yr av. salary) | Intravenous Equipment | Computer Hardware (amortized over 3 yrs) | Computer Software (amortized over 3 yrs) | Total Cost | Cost Per Student |
|---|---|---|---|---|---|---|
| Computer Learners (N=48) | $8.64<br>(60 minutes) | $10.42<br>(2 sets) | $24.99<br>(5 TRS-80 Model I's used 1 week out of 28) | $25.00<br>(5 copies of computer program on tape) | $69.05 | $1.44 |
| Laboratory Learners (N=48) | $54.72<br>(380 minutes) | $41.68<br>(8 sets) | ---------- | ---------- | $96.40 | $2.01 |

## DISCUSSION

The findings of this study demonstrate that a computer simulation is an effective, efficient, and economical methodology for teaching nursing students to calculate and regulate IV flow rates, a basic nursing skill with both cognitive and psychomotor components. These findings have implications for nursing education, both in educational institutions and in health care agencies, and for patients themselves. Specific implications are for the potential use of CAI to: (1) teach other basic nursing psychomotor and cognitive skills; (2) teach complex nursing skills, such as clinical problem-solving; (3) validate prior knowledge of non-traditional nursing students; (4) "refresh" and update knowledge and clinical skills of current and returning practitioners; and (5) teach patients and their families to monitor and provide for their own health care needs. Additional implications of this study concern the potential for microcomputers, with their many advantages and special capabilities, to upgrade the quality and individualization of nursing education.

## REFERENCES

Bitzer, M., & Bitzer, D. Teaching nursing by computer: an evaluative study. *Computers in Biology and Medicine*, 1973, *3*, 187-204.

Bitzer, M., Boudreaux, M., & Avner, R.A. *Computer-based instruction of basic nursing utilizing inquiry approach*. Urbana, Illinois: Computer-based Education Research Laboratory, February, 1973.

Campbell, D.T., & Stanley, J.C. Experimental and quasi-experimental designs for research on teaching (Chapter 5), in N.L. Gage (Ed.), *Handbook of research on teaching*. Chicago: Rand McNalley & Company, 1963.

Droste-Bielak, E. *The use of computer simulation in teaching interviewing techniques to beginning nursing students in a baccalaureate program*. Unpublished doctoral dissertation, University of Michigan, 1980.

Huckabay, L., Anderson, N., Holm, D., & Lee, J. Cognitive, affective, and transfer of learning consequences of computer-assisted instruction. *Nursing Research*, July-August, 1979, *28*, 228-233.

Kirchhoff, K.T., & Holzemer, W. Student learning and a computer-assisted instruction program. *Journal of Nursing Education*, March, 1979, *18*, 22-30.

Rubinson, L.G., & Robinson, J.A. Computer-assisted instruction in health education: a pilot study. *International Journal of Instructional Media*, 1977-1978.

Valish, A.U., & Boyd, N.J. The role of computer-assisted instruction in continuing education of registered nurses: an experimental study. *Journal of Continuing Education in Nursing*, January-February, 1975, *6*, 13-32.

# THE USE OF NURSING PROCESS IN COMPUTER-BASED SIMULATION DESIGN

KATHLEEN D. KADNER

THE UNIVERSITY OF TEXAS AT EL PASO

## ABSTRACT

Computer-based simulations (CBS) are currently being developed for use in nursing education. These simulations are generally regarded as attempts to provide practice in decision-making, applying didactic material already covered through more traditional teaching strategies. Development of computer-based nursing simulations is being facilitated at the University of Texas at El Paso College of Nursing and Allied Health by the use of a nursing process model for module design. This paper describes the rationale and use of the nursing process as a model for the developmental design of computer-based nursing clinical simulations. Nursing process was decided upon because it offered a well-established decision-making tool for clinical nursing practice; a coherent structure for the simulation; and facilitation of the authoring process by subject matter experts.

## NURSING PROCESS

Nursing process is a decision-making tool for clinical nursing. An outgrowth of the scientific method, nursing process provides a means of establishing a rationale and strategies for nursing care. Several authors have described this process in detail (Sundeen, Gail, Ramkin, and Cohen 1976, Yura and Walsh 1978, McCarthy 1981) using different nomenclature, but key elements common to all are the systematic identification and action strategies which must be undertaken for effective nursing care to occur. These include data collection, assessment, planning, implementation, and evaluation. As defined by Bregg and Reed (1972), nursing process is a set of intellectual tasks and cognitive actions in a special order which is initiated when the nurse and patient or patients come together and the nurse recognizes the need for purposeful interaction.

## COMPUTER-BASED CLINICAL SIMULATIONS

Computer-based simulations in nursing involve the learner as participant in a situation in which some informed decisions must be made effecting clinical outcomes. The situation is presented textually and graphically projected to the learner's vision on a CRT screen and the student interacts by means of a keyboard, set up similarly to a typewriter keyboard. Touch sensitive screens, light pens for response input, voice activation are technological developments which are available, but the most common mode of learner/computer interaction is keyboard/CRT display. The six computers currently in the use at the College are microcomputers: Intelligent Systems Corporation's Intecolor 3600 and Apple II, equipped with 64K of memory.

## SIMULATION AS A TEACHING STRATEGY

The use of non-computerized simulation as a teaching strategy in nursing education is already regarded as a prerequisite to bona fide clinical assignment. Nursing simulation is the provision of life-like problem solving experiences related to present or future clinical practice. The advantages of simulation are: no risk or harm to patients; practice in clinical skills; active learning experience; opportunity for cognitive-behavioral change. With the immediate feedback that is a necessary component of computerized instruction, the learner additionally can recognize himself as a causative agent in the experience. A nursing computerized simulation's primary educational value is in providing the opportunity for applying theory to practice. Clinical decision-making practice facilitates transfer of learning to real life clinical situations (Huckabay, Anderson, Holm, and Lee 1979). The factual material imparted by the simulation is incidental to this learning objective.

## NURSING PROCESS LINK TO CBS

Computerized clinical simulations are being used in the education of other health care professionals. Medical computer simulations have generally revolved about the formulation of diagnostic and treatment strategies. Nursing, while also involved with medical diagnosis and treatment, is primarily concerned in the clinical setting with surveillance. The nurse is often the only individual physically present with patients in a clinical setting on a 24 hour basis and is often the first person in a position to notice changes in behavior, symptomatology, etc. Since these changes must be handled in a rational manner --> enter nursing process. The nurse should be able to assess a situation and operate within the constraints of relevant facts and available options. To perform in this manner, the nurse should be expert at decision-making so that the situation can be managed with minimum risk to the patient's well being.

Nursing process can be recognized as lending coherence to the structure of the simulation. The learner gains practice in making clinical decisions in an orderly fashion which should parallel real-life practice. The steps in the nursing process can be summarized as data collection, planning, implementation, and evaluation. Typically, simulations are designed to take place during major clinical changes in a patient. These events can be initiated, worked on and completed in the "real time" that the learner is at the computer terminal. Additionally, in a crisis several decisions must be made quickly and in the correct sequence, so that the immediacy and importance of the learner's actions are more apparent. However, simulations can cover any aspect of health care -- from preventive to rehabilitative -- over any time span. The learner is simply informed of the time period during which events are occurring.

## DESCRIPTION OF PROJECT

At the University of Texas at El Paso College of Nursing and Allied Health, a D.H.H.S. grant project entitled "Safeguarding Patients Via Tri-Simulation in Nursing" has been in progress since 1979. One aspect of this project was the establishment of a Microcomputer Laboratory. One of the purposes of this lab has been to explore microcomputer applications in nursing education, especially computer-based simulation. The computer is interfaced to a life-sized mannikin, which simulates the pathophysiological changes that the computer-simulated patient is experiencing. The computer-based simulation design development included finding a format which would have particular relevance to nursing and provide a familiar structure for nursing subject matter experts to author in. Of course, the bottom line in design development is whether the format chosen will best serve the instructional objectives. Nursing process is an inductive approach to clinical problem solving; a nursing computerized simulation's major purpose is to provide practice in clinical problem solving.

Although nursing process has provided a satisfactory design approach for the clinical simulations which have been developed, there are undoubtedly many other effective formats for designing nursing computer-based simulations. This module design has enabled authors to delineate their material more easily than a random approach might, but it must be stressed that there is no need to "lock into" this particular developmental design.

## FORMAT FOR COMPUTER-BASED NURSING CLINICAL SIMULATIONS

Introduction: Welcome, Objectives, Expectations, Optional Instructions

Scenario: Situation (time and place referenced)

Data Collection: Observation, Written Record, Interview, Lab and X-ray results, Physical Assessment

Planning, Priorities, Goals: Nursing diagnosis, Low level vs. high level priorities

Implementation: Psycho-social interactions, health teaching, procedures, monitoring, referal, medication administration, discharge planning, emergency measures

Evaluation: Effectiveness of data collection, planning, and interventions in achieving nursing care goals.

## APPLICATION OF NURSING PROCESS TO CBS

In the data collection segment, the learner has access to a written record (chart, work record, etc.), lab and x-ray results. The written record, lab tests and results are available as a print-out on student's command: "If you would like a printed copy of this information, type 'PRINT' and press the return/enter key." An interview and physical assessment (complete, partial or not necessary) could be conducted by the learner. The interview section is designed as an inquiry from student to patient. For example, the learner would inquire "When did your pains start?" in English or using a code number/question. The patient responds in English.

Prompts with rationales from the author are ongoing. For example, if the student wants to exit the data collection segment before a physical assessment, and the author has determined that a physical assessment is necessary for this case, the student may be prevented from going on "You'll need to do a physical assessment before you can decide on priorities for this patient because...". Or "A complete physical assessment would be inappropriate at this time, NAME, because emergency measures should be instituted immediately." The partial physical assessment consists of a list of assessment criteria, temperature, pulse, neurologic signs, etc...from which the student can choose. An exit from the partial physical assessment section can be made whenever the student believes that enough information has been collected. A complete physical assessment may be handled as a text readout.

In the priority/goal setting section, relevant information is extracted from prior data collection by the learner to formulate a nursing diagnosis and develop a nursing care plan with high and low level priority actions. Determination of the relevance of information obtained in the data collection segment is made.

In the implementation section the student is presented with a menu of possible interventions divided into categories. Possible categories are psychosocial interactions, health teaching, procedures, monitoring, referral, medication administration, discharge planning, emergency measures. Each category chosen might have a set of multiple choice options or a list of possible strategies that the student can choose from. For example, under psychosocial interactions, the choices might be: 1. Contact patient's family. 2. Use Rogerian reflective technique to focus more on this patient's anxiety. 3. Confront this patient with the irrationality of her fears. 4. Introduce the patient to an ambulatory patient who has undergone the same surgery.

Feedback would include reinforcement for correct decisions and branching to a new set of circumstances resulting from that action. If incorrect, the learner may be looped back to the original set of options or allowed to continue with a less than optimal set of consequences resulting from a poor decision. Branching is limited only by the imagination and stamina of the author(s) and instructional programmer(s). If the simulation involves an emergency in which life-saving measures must be instituted in a rather well-defined, rapid sequence, the intervention segment would necessarily differ from a non-emergency situation.

Evaluation is ongoing throughout the simulation in the feedback given with each response. An optional case discussion branch provides additional decision evaluation.

## CONCLUSION

The development of computer-based clinical nursing simulations has been enhanced at the University of Texas at El Paso College of Nursing and Allied Health by the use of the nursing process. When this format is suggested to faculty authors, there is immediate conceptual recognition. This familiarity with a prospective design simplifies the coordination process, since at least one aspect of the prospective program--its structure--does not present additional learning for the author. Nursing process is the means by which the nursing practitioner performs in the clinical setting. Using nursing process for simulations thus accomplishes achievement of learning objectives and facilitation of nursing faculty authorship.

## REFERENCES

Bregg, E. and Reed, S. The Nursing Process. System-Wide Curriculum Committee Materials. University of Texas (San Antonio). July, 1972.

Huckabay, L.M.D., Anderson, N., Holm, D.M., and Lee, J. Cognitive, affective, and transfer of learning consequences of computer-assisted instruction. Nursing Research. July/August, 1979. Vol. 28, No. 4. pp. 228-233.

McCarthy, M.M. Nursing process: application of current thinking in clinical problem solving. Journal of Advanced Nursing. May, 1981. Vol. 6. pp. 173-7.

Sundeen, S.J., Gail, W., Ramkin, E.D., and Cohen, S.P. Nurse-Client Interaction. St. Louis: C. V. Mosby Co. 1976. pp. 14-18.

Yura, H. and Walsh, M. The Nursing Process. New York: Appleton-Century-Crofts. 1978. pp. 26-37.

# CAI AND INSTRUCTIONAL INTERACTIONS

Louis A. Pagliaro, M.S., Pharm.D., Ph.D.

University of Alberta

## ABSTRACT

This research project was designed to document the occurrence of significant instructional interactions which had previously been reported in the literature to involve the use of CAI. The subjects for this study were pharmacy students enrolled in an undergraduate pharmacology course at the University of Alberta during the 1979-1980 academic year. The subjects were randomly divided into three different treatment groups (CAI alone, CAI concurrent with lecture, lecture alone) and proceeded through a series of CAI programs in a variety of topics in pharmacology. Data was collected on a variety of demographic and personologic variables and on academic performance in a selected pharmacology topic. Several instructional interactions previously reported in the literature to be related to CAI were observed and discussed. None of these instructional interactions appear to be unique to CAI per se. Thus, a description of CAI as an essentially neutral technology, which has the potential to interact within the educational milieu in either a positive or negative manner, seems to have been supported by the findings of this study.

## Procedures and Methodology

This project was designed to examine and answer the following question.

Can a significant instructional interaction be found between specific student demographic or personologic variables and academic performance in relation to the use of CAI in pharmacology?

Subjects. The subjects for this study were pharmacy students enrolled in an undergraduate pharmacology course at the University of Alberta during the 1979-1981 academic years. The subjects were randomly divided into three different treatment groups (CAI alone, CAI concurrent with lecture, and lecture alone) and proceeded through a series of CAI programs in a variety of topics in pharmacology. Data was collected on a variety of demographic and personologic variables and on academic performance in a selected pharmacology topic.

Procedure. Analysis of variance techniques were used to determine if particular learner variables interacted significantly with CAI to differentially produce academic achievement increments or decrements. Each of these analyses examined performance on pretest X and posttest X #1 of students possessing particular demographic or personologic variables in relation to treatment group (i.e. TI-CAI alone, T2-CAI concurrent with lecture, T3-lecture alone). The variables selected were those found in the literature which had previously yielded a significant correlation or instructional interaction in relation to academic achievement (See Table 1). These variables included: achievement motivation, aptitude, sex and sociability. The variables of anxiety and mental ability, although listed as possibly significant in Table 1, were not used for the following reasons. Measurement of anxiety was attempted by means of an anxiety questionnaire. However, when the attitude questionnaire was factor analyzed, the scales failed to load significantly upon any one factor. Thus, a valid measure of anxiety which could be used to divide subjects into low, medium, and high groups for analysis was not available. Normed tests, which could measure mental ability in relation to the subject matter of pharmacology, were also not readily available, therefore, this variable was not measured. However, a somewhat related variable "pharmacology knowledge entry level" was measured and analyzed for the presence of instructional interactions.

## Results

Achievement Motivation. Achievement motivation was the first variable used to define a factor in a two-way analysis of variance. Achievement motivation was measured by the Achievement via Independence (Ai) scale of the California Psychological Inventory (CPI). Subjects were divided on the basis of their Ai scores on CPI #1 into three groups: low, Ai=12-16; medium, Ai=17-19; and high, Ai=21-25.

Table 2 lists a summary analysis of variance table for academic achievement pretest X in relation to level of achievement motivation and treatment. Neither the 'achievement motivation' nor 'treatment' main effects were significant. The 'achievement motivation X [by] treatment' interaction was also not significant. Similar analyses for posttest X #1 are presented in Table 3. None of these effects were statistically significant.

Thus, it appears that the results of this study fail to demonstrate an instructional interaction between academic achievement and achievement motivation, as measured by the Ai scale of the CPI. This result is in agreement with that which was reported by Reid, et al (1973).

Aptitude. The summary three-way analysis of variance table for the next variable, aptitude, is found in Table 4. Aptitude was measured by grade point average (GPA) and divided into three groups: low, GPA=4.38-5.50; medium, GPA=5.61-6.84; and high, GPA=6.89-8.76. These GPA's were based on cummulative performance through the first two years in the pharmacy program at the University of Alberta. The GPA's are theoretically based on a stanine (marking on a 1-9) system, however, a minimum GPA of 4.0 is necessary to remain and progress within the pharmacy program. Thus, no GPA's less than 4.0 were found in this study's sample.

The 'aptitude' main effects and the 'occasion (0,1)' (i.e. pretest X to posttest X #1) main effects were statistically significant and post-hoc contrasts for these effects are presented in Table 5. The remaining analyses listed in Table 4 were not significant.

From Table 5 it is apparent that aptitude, as determined by GPA, does not significantly interact in a differential manner with academic achievement (i.e. all groups benefited equally). This finding is contrary to the two previously reported studies concerning aptitude and academic achievement (Deigman, et al, 1980; Reid, et al, 1973) which found an interaction when CAI was used. However, as noted above, a minimum GPA was required to stay in the pharmacy program. Thus, the distribution of GPA's used in this study were higher than the true population of GPA's and was skewed to the right. This feature may have affected the analyses and may account for the discrepancy in findings from the previously reported studies.

Pharmacology knowledge entry level. The next variable analyzed for instructional interactions was that of pharmacology knowledge entry level. The pretest X, which the students completed at the beginning of the experimental phase of this project, served as an entry level measure of "pharmacologic" knowledge. Accordingly, the students were divided into three groups on the basis of their performance: low, number of correct responses on pretest X=1-3; medium, number of correct responses on pretest X=4-5; and high, number of correct responses on pretest X=6-8.

Table 7 contains the summary analysis of variance table for academic achievement pretest X in relation to pharmacology knowledge entry level and treatment group. The 'pharmacology knowledge' main effects were significant, but the 'treatment' main effects were not. The finding of significant 'pharmacology knowledge' effects would be expected because the analysis was deliberately set up to have different levels of pretest achievement. Thus, this analysis confirmed that the three groups, which had been divided according to the number of correct responses on pretest X, were statistically different. Table 8 presents the same analysis as Table 7 only in relation to performance on posttest X #1. No significant effects are noted in Table 8. However, in this case, one would have expected the 'pharmacology knowledge' main effects to have remained significant. Since they did not, this indicates a significant interaction has occurred between pharmacology knowledge entry level and academic achievement.

If one were to plot the mean pretest X and posttest X #1 scores presented in Table 6 and examine the resultant graphs, one would find that those students in the low and medium pretest X categories gained most from all three treatments, while those in the high category gained the least. In fact, those in the high category appeared to show almost no gain from pretest X to posttest X #1. This effect occurred across all three treatments. This was not a ceiling effect because the maximum possible score on the achievement tests was 15.

This finding is believed to be the first such reported effect in relation to CAI use in the content area of pharmacology. However, if pharmacology knowledge entry level is considered as a measure of mental ability in the content area of pharmacology then this finding would be in agreement with the three previously published reports (Edwards, et al, 1975; Lavin, 1980; Suppes & Morningstar, 1969) which all found evidence of instructional interactions of mental ability with academic achievement when using CAI. In this regard, Tobias (1976) has noted that:

> Level of prior achievement can, of course, be easily defined by pretest scores . . . the higher the level of prior achievement, the lower the instructional support required to accomplish instruction objectives. Conversely, as the level of prior achievement decreases, the amount of instructional support required increases. (p. 67)

However, noting that the observed interaction persisted across treatments, it is probably a more likely conclusion that the interaction is with the design of both the CAI and lecture teaching strategies and not with CAI per se. In other words, the lectures and CAI programs were probably designed (consciously or subconsciously) in a norm referenced manner which attempted to bring all students to a certain desired minimal level of achievement.

Sex. The variable of sex was the next one analyzed for instructional interactions. Subjects were divided into two groups, male and female, for this analysis (See Table 9).

The summary two-way analysis of variance table for academic achievement pretest X in relation to treatment group and sex is presented in Table 10. The 'treatment' main effects were not significant, however, the 'sex' main effects were significant. This indicated that females had significantly lower pretest X mean scores than did males. Table 11 presents a similar analysis only in relation to posttest X #1. Here we note that none of the effects were significant (i.e. mean scores for females and males were now statistically equal).

This implies that the females had significantly greater posttest X #1 performance increases in relation to their pretest X scores, which had been significantly lower than the males. Thus, it appears that this study suggests the presence of an instructional interaction with females achieving relatively greater academic achievement than males. It further appears that this finding is in conflict with the previous two studies (Downing & Lowe, 1981; Wooley, 1978) which also found an apparent instructional interaction with sex, but with males achieving preferentially higher than females.

An instructional interaction probably has occurred, not between sex and CAI, but between sex and the subject matter. In the previously reported studies, males did relatively better in the "mathematically' oriented subjects. In this study females did relatively better in the "verbally" oriented topic of pharmacology This interpretation is borne out by the observation that the observed instructional interaction with sex occurred across all three experimental treatments (i.e. CAI alone, CAI concurrent with lecture, lecture alone).

Sociability. The last variable analyzed for instructional interactions was that of sociability. Sociability was measured by the Sociability (Sy) scale of the CPI. Subjects were divided into three groups on the basis of their Sy scores on CPI #1: low, Sy=15-20 (with one observation at 8); medium, Sy=21-25; and high, Sy=26-34 (See Table 12).

Table 13 lists the summary analysis of variance table for academic achievement pretest X in relation to level of sociability and treatment group. None of the effects noted in Table 13 were significant. Similar analyses for posttest X #1 are found in Table 14. Here we note that the 'treatment' and the 'sociability X treatment' effects were not significant but that the 'sociability' main effects were significant. Table 12 indicates that the posttest X #1 scores for the low sociability group were higher than those obtained by either the medium or high sociability groups across all treatments. Post hoc contrasts further indicates that the low sociability group achieved significantly higher mean scores than the medium sociability group on the posttest X #1 performance.

Thus, this study has demonstrated an instructional interaction between level of sociability and academic achievement. All three experimental treatments noted the same effect and all three were primarily oriented toward independent study. This result demonstrates that the results observed by Reid, et al (1973), were not with CAI, but with the design of the CAI experience. Thus, if students had been assigned to work at their CAI experience in pairs, instead of alone, the observed interaction would be expected to have been the opposite, as noted by Sutter and Reid (1969).

## Conclusion

A number of instructional interactions have been demonstrated in this study. However, in each case, the interaction has persisted across treatments. Thus, it appears that instructional interactions, which in the past had been tentatively associated with a specific treatment (i.e. CAI), may have actually been related to more basic variables such as the subject matter being taught or elements of the instructional process which in the present study were common to all three treatments.

Thus, although several instructional interactions were observed in the context of CAI, none of these appear to be unique to CAI per se. A description of CAI as an essentially neutral technology, which has the potential to interact within the educational milieu in either a positive or negative manner, seems to have been supported by the findings of this study.

## Acknowledgements

I would like to thank Ann M. Pagliaro and Stephen M. Hunka for their assistance in this project.

This project was supported in part by Alberta Advanced Education and Manpower.

## References

Deigman, G.M., Seager, B.R., Kimball, M. & Horozitz, N.S. Computer-assisted, programmed text, and lecture modes of instruction in three medical training courses: comparative evaluation (AFHRL-TR-79-76). Lowry Air Force Base, Colorado: Air Force Human Resources Laboratory, Technical Training Division June 1980. (ERIC Document Reproduction Service No. ED 192 717)

Downing, B.D., & Lowe, N.W. Mathematics anxiety and performance with CAI. ADCIS NEW, 1981, 14, 5 (Abstract)

Edwards, J., Norton, S., Taylor, S., Weiss, M., & Dusseldorp, R. How effective is CAI? A review of the research. Educational Leadership, 1975, pp. 147-153.

Fletcher, J.D., & Atkinson, R.C. Evaluation of the Stanford CAI program in initial reading. Journal of Educational Psychology, 1972, 63, 597-602.

Lavin, B.F. Can Computer-assisted instruction make a difference? Teaching Sociology, 1980, 7, 163-179.

Leherissey, B.L., O'Neil, H.F., Henreich, D.L., & Hansen, D.N. Effect of anxiety response mode, subject matter familiarity, and program length on achievement in computer-assisted learning. Journal of Educational Psychology, 1973, 64, 310-324.

Pagliaro, L.A. Instructional interactions and the mega interactive model of instruction. Educational Technology, 1979, 19, 35-39.

Pagliaro, L.A. Analysis of computer assisted instruction in pharmacology. Proceedings of the Western Pharmacology Society, 1981, 24, 113-115.

Pagliaro, L.A., Bidwell, C.M., Cook, D.A., & Heath, C. CAI in pharmacology at the University of Alberta. Project report presented at the Association for the development of Computer-based Instructional Systems (ADCIS), San Diego, California, February 1979. (ERIC Document Reproduction Service No. 175 477)

Pagliaro, L.A., & Burkhalter, A. Application and testing of computer aided instruction in pharmacology. Paper presented at the Association for the Development of Computer-based Instructional Systems (ADCIS), San Diego, California, February 1979. (ERIC Document Reproduction Service No. ED 175 477)

Rappaport, E. Effects of dogmatism and anxiety during computer-assisted learning. Psychological Reports, 1975, 37, 1055-1065.

Reid, J.B., Palmer, R.L., Whitlock, J., & Jones, J. Computer-assisted instruction performance of student pairs as related to individual differences. Journal of Educational Psychology, 1973, 65, 65-73.

Rokeach, M. Political and religious dogmatism: an alternative to the authoritarian personality. Psychological Monographs: General and Applied, 1956, 70, 1-43.

Sarason, I.G. Intellectual and personality correlates of text anxiety. Journal of Abnormal and Social Psychology, 1959, 59, 272-275.

Spielburger, C.D., O'Neil, H.F., & Hansen, D.N. Anxiety, drive theory, and computer-assisted learning. In B.A. Mahler (Ed.), Progress in experimental personality research (Vol. 6), New York: Academic Press, 1972.

Suppes, P., & Morningstar, M. Computer-assisted instruction. Science, 1969, 166, 343-350.

Sutter, E.G., & Reid, J.B. Learner variables and interpersonal conditions in computer-assisted instruction. Journal of Educational Psychology, 1969 60, 153-157.

Tobias, S. Distraction, response mode, anxiety, and achievement in computer-assisted instruction. Journal of Educational Psychology, 1973, 65, 233-237.

Wooley, J.K. Factors affecting student's attitudes and achievement in an astronomy computer-assisted instructional program. Journal of Research in Science Teaching, 1978, 15, 173-178.

Table 1. Summary of "CAI and Instructional Interactions" Studies

| Variable | Measure | Observation[1] | Author(s) |
|---|---|---|---|
| Achievement Motivation | Achievement via Independence Scale of the CPI[2] | - | Reid, et al, 1973 |
|  | Armed Services Tests[3] | + | Deigman, et al, 1980 |
| Anxiety | Sarason[4] Scale | + | Sutter & Reid, 1969 |
|  | STAI[5] A - State Scale | + | Spielberger, et al, 1972 |
|  | STAI A - Trait | + | Leherissey, et al, 1973 |
|  | STAI A - State Scale | - | Tobias, 1973 |
|  | Sarason Scale | - | Reid, et al, 1973 |
|  | STAI A - State Scale | + | Rappaport, 1975 |
|  | Test of Mathematics anxiety | + | Downing & Lowe, 1981 |
| Aptitude | SAT - M[6] | + | Reid, et al, 1973 |
|  | Armed Services Tests[7] | + | Deigman, et al, 1980 |
| Attitude | CAI - Math Attitude Survey | - | Reid, et al, 1973 |
| Dogmatism | Rokeach[8] Dogmatism Scale | - | Rappaport, 1975 |
| Dominance | Dominance Scale of the CPI[9] | - | Sutter & Reid, 1969 |
|  | Dominance Scale of the CPI | - | Reid, et al, 1973 |
| Flexibility | Flexibility Scale of the CPI | - | Reid, et al, 1973 |
| Mental Ability | Stanford Achievement Tests | + | Suppes & Morningstar, 1969 |
|  | ________ | + | Edwards, Et al, 1975 |
|  | Grade on Lecture Test | + | Lavin, 1980 |

(Table 1 continued)

| | | | |
|---|---|---|---|
| Sex | Male or Female | - | Fletcher & Atkinson, 1972 |
| | Male or Female | - | Reid, et al, 1973 |
| | Male or Female | + | Wooley, 1978 |
| | Male or Female | + | Downing & Lowe, 1981 |
| Sociability | Sociability Scale of the CPI | + | Sutter & Reid, 1969 |
| | Sociability Scale of the CPI[10] | - | Reid, et al, 1973 |

---

[1] "+" signifies a significant correlation or instructional interaction in relation to achievement.
"-" signifies lack of significant correlation or instructional interation in relation to achievement.

[2] Achievement via Independence (Ai) scale of the California Psychological Inventory.

[3] These tests specifically measured "level of aspiration to achieve".

[4] Sarason, I.G. Intellectual and personality correlates of test anxiety. *Journal of Abnormal and Social Psychology*, 1959, *59*, 272-275.

[5] State-trait anxiety inventory.

[6] Mathematics section of the scholastic aptitude test.

[7] These tests consisted of the "Armed Services Vocational Aptitude Battery" and the "Delta Training Aptitude Battery".

[8] Rokeach, M. Political and religious dogmatism: an alternative to the authoritarian personality. *Psychological Monographs*, 1956, *70*, 1-43.

[9] Delta-training aptitude battery.

[10] In this CAI study students were assigned to work in pairs.

Table 2. Summary two way analysis of variance table for academic achievement pretest X in relation to level of achievement motivation and treatment group

Summary Analysis of Variance Table

| Source of Variation | DF | MS | F | P |
|---|---|---|---|---|
| Achievement motivation main effects | 2 | 2.13 | 0.63 | 0.54 |
| Treatment main effects | 2 | 0.57 | 0.17 | 0.85 |
| Achievement motivation X Treatment Interaction | 4 | 2.07 | 0.61 | 0.66 |
| Error | 52 | 3.38 | | |

n = 61

Table 3. Summary two way analysis of variance table for academic achievement posttest X #1 in relation to level of achievement motivation and treatment group

Summary Analysis of Variance Table

| Source of Variation | DF | MS | F | P |
|---|---|---|---|---|
| Achievement motivation main effects | 2 | 2.08 | 0.51 | 0.61 |
| Treatment main effects | 2 | 1.87 | 0.46 | 0.64 |
| Achievement motivation X Treatment Interaction | 4 | 1.63 | 0.40 | 0.81 |
| Error | 52 | 4.09 | | |

n = 61

Table 4. Summary three way analysis of variance table for aptitude, treatment group, and academic achievement

Summary Analysis of Variance Table

| Source of Variation | DF | MS | F | P |
|---|---|---|---|---|
| Between subjects | 53 | | | |
| Aptitude main effects | 2 | 16.29 | 4.11 | 0.02* |
| Treatment main effects | 2 | 0.84 | 0.21 | 0.81 |
| Aptitude X Treatment Interaction | 4 | 2.06 | 0.52 | 0.72 |
| Subjects within groups | 45 | 3.96 | | |
| Within Subjects | 54 | | | |
| Occasion (0,1) main effects | 1 | 98.23 | 34.07 | 0.00* |
| Aptitude X Occasion (0,1) interaction | 2 | 0.40 | 0.14 | 0.87 |
| Treatment X Occasion (0,1) interaction | 2 | 1.45 | 0.50 | 0.61 |
| Aptitude X Treatment X Occasion (0,1) interaction | 4 | 1.70 | 0.59 | 0.67 |
| Occasion (0,1) X subjects within groups | 45 | 2.88 | | |

n = 54
*significant at $p<0.05$

Table 5. Post-hoc contrasts for the three way analysis of variance among levels of academic achievement for given levels of aptitude and treatment group.

| | Pretest X[1] | Posttest X #1[1] | DF1 | DF2 | F | P |
|---|---|---|---|---|---|---|
| Aptitude | | | | | | |
| Low GPA | 4.00 | 5.67 | 1 | 45 | 5.78 | 0.02* |
| Medium GPA | 3.61 | 5.67 | 1 | 45 | 8.79 | 0.01* |
| High GPA | 4.89 | 6.89 | 1 | 45 | 8.32 | 0.01* |
| Treatment | | | | | | |
| T1 | 3.89 | 6.01 | 1 | 45 | 9.77 | 0.00* |
| T2 | 4.22 | 6.33 | 1 | 45 | 9.27 | 0.00* |
| T3 | 4.39 | 5.83 | 1 | 45 | 4.34 | 0.04* |

[1] Mean scores
n = 54
*significant at $p<0.05$

Table 6. Mean scores[1] [2] and standard deviations obtained from the academic achievement tests presented according to pharmacology knowledge entry levels by treatments.

| | | Pretest X | | | | Posttest X #1 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | T1 | T2 | T3 | | T1 | T2 | T3 | |
| LEVEL[3] | Low | 2.33<br>±0.71<br>(n=9) | 2.86<br>±0.38<br>(n=7) | 2.44<br>±0.73<br>(n=9) | [2.54] | 5.56<br>±1.33<br>(n=9) | 5.29<br>±1.70<br>(n=7) | 5.11<br>±2.57<br>(n=9) | [5.32] |
| | Medium | 4.33<br>±0.52<br>(n=6) | 4.30<br>±0.48<br>(n=10) | 4.50<br>±0.55<br>(n=6) | [4.38] | 6.00<br>±1.26<br>(n=6) | 7.00<br>±1.76<br>(n=10) | 6.33<br>±1.86<br>(n=6) | [6.44] |
| | High | 6.40<br>±0.89<br>(n=5) | 7.00<br>±1.41<br>(n=4 | 6.80<br>±1.10<br>(n=5) | [6.73] | 6.20<br>±2.86<br>(n=5) | 6.75<br>±2.06<br>(n=4) | 6.80<br>±1.92<br>(n=5) | [6.58] |
| | | [4.36] | [4.72] | [4.58] | | [5.92] | [6.35] | [6.08] | |

(Table 6 continued)

[1] The values in brackets, which are underlined, are the mean scores for ability groups summed over treatments.

[2] The values in brackets, which are not underlined, are the mean scores for treatments summed over pharmacology knowledge entry level groups.

[3] Level refers to pharmacology knowledge entry level as determined by pretest X scores.

Table 7. Summary two-way analysis of variance table for academic achievement pretest X in relation to pharmacology knowledge entry level and treatment group.

Summary Analysis of Variance Table

| Source of Variation | DF | MS | F | p |
|---|---|---|---|---|
| Pharmacology knowledge main effects | 2 | 78.49 | 145.68 | 0.00* |
| Treatment main effects | 2 | 0.63 | 1.16 | 0.32 |
| Pharmacology knowledge X Treatment interaction | 4 | 0.29 | 0.54 | 0.71 |
| Error | 52 | 0.54 | | |

n = 61

*significant at $p<0.05$

Table 8. Summary two way analysis of variance table for academic achievement posttest X #1 in relation to pharmacology knowledge entry level and treatment group.

Summary Analysis of Variance Table

| Source of Variation | DF | MS | F | p |
|---|---|---|---|---|
| Pharmacology knowledge main effects | 2 | 10.45 | 2.74 | 0.07 |
| Treatment main effects | 2 | 0.96 | 0.25 | 0.78 |
| Pharmacology knowledge X Treatment interaction | 4 | 1.05 | 0.28 | 0.89 |
| Error | 52 | 3.81 | | |

n = 61

Table 9. Mean scores[1] [2] and standard deviations obtained from the academic achievement tests presented according to treatments by sex.

| | | Pretest X | | | Posttest X #1 | | |
|---|---|---|---|---|---|---|---|
| | | Male | Female | | Male | Female | |
| TREATMENTS | T1 | 4.60<br>±2.41<br>(n=5) | 3.73<br>±1.62<br>(n=15) | [4.17] | 6.80<br>±2.49<br>(n=5) | 5.53<br>±1.36<br>(n=15) | [6.17] |
| | T2 | 5.57<br>±2.15<br>(n=7) | 3.71<br>±0.83<br>(n=14) | [4.64] | 6.43<br>±1.90<br>(n=7) | 6.36<br>±1.95<br>(n=14) | [6.39] |
| | T3 | 4.33<br>±2.18<br>(n=9) | 4.00<br>±1.84<br>(n=11) | [4.17] | 6.44<br>±2.60<br>(n=9) | 5.46<br>±1.92<br>(n=11) | [5.95] |
| | | [4.83] | [3.82] | | [6.58] | [5.78] | |

[1] The values in brackets, which are underlined, are the mean scores for treatment groups summed over sexes.

[2] The values in brackets, which are not underlined, are the mean scores summed over treatments.

Table 10. Summary two way analysis of variance table for academic achievement pretest X in relation to treatment group and sex.

Summary Analysis of Variance Table

| Source of Variation | DF | MS | F | p |
|---|---|---|---|---|
| Treatment main effects | 2 | 1.38 | 0.45 | 0.64 |
| Sex main effects | 1 | 13.68 | 4.46 | 0.04* |
| Treatment X Sex interaction | 2 | 2.85 | 0.93 | 0.40 |
| Error | 55 | 3.07 | | |

n = 61
*significant at $p<0.05$

Table 11. Summary two way analysis of variance table for academic achievement posttest X #1 in relation to treatment group and sex.

Summary Analysis of Variance Table

| Source of Variation | DF | MS | F | p |
|---|---|---|---|---|
| Treatment main effects | 2 | 0.94 | 0.24 | 0.78 |
| Sex main effects | 1 | 7.94 | 2.05 | 0.16 |
| Treatment X Sex interaction | 2 | 1.72 | 0.44 | 0.64 |
| Error | 55 | 3.86 | | |

n = 61

Table 12. Mean scores [1] [2] and standard deviations obtained from the academic achievement tests presented according to level of sociability (Sy) by treatments

| | | Pretest X | | | | Posttest X #1 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | T1[3] | T2 | T3 | | T1 | T2 | T3 | |
| S<br>O<br>C<br>I | Low | 4.17<br>±2.56<br>(n=6) | 4.29<br>±1.50<br>(n=7) | 4.17<br>±1.47<br>(n=6) | [4.21] | 6.17<br>±2.48<br>(n=6) | 7.57<br>±1.27<br>(n=7) | 7.17<br>±2.04<br>(n=6) | [6.97] |
| A<br>B<br>I<br>L | Medium | 4.33<br>±1.63<br>(n=6) | 3.57<br>±0.79<br>(n=7) | 4.30<br>±1.94<br>(n=10) | [4.07] | 5.83<br>±1.17<br>(n=6) | 5.57<br>±2.30<br>(n=7) | 5.00<br>±2.26<br>(n=10) | [5.47] |
| I<br>T<br>Y | High | 3.50<br>±1.41<br>(n=8) | 5.14<br>±2.12<br>(n=7) | 3.75<br>±2.99<br>(n=4) | [4.13] | 5.63<br>±1.60<br>(n=8) | 6.00<br>±1.57<br>(n=7 | 6.25<br>±1.89<br>(n=4) | [5.96] |
| | | [4.00] | [4.33] | [4.07] | | [5.88] | [6.38] | [6.14] | |

---

[1] The values in brackets, which are underlined, are the mean scores for level of socialization summed over treatments.

[2] The values in brackets, which are not underlined, are the mean scores for treatments summed over sociability levels.

[3] T1 is CAI alone; T2 is CAI concurrent with lecture; and T3 is lecture alone.

Table 13. Summary two way analysis of variance table for academic achievement pretest X in relation to level of sociability and treatment group.

Summary Analysis of Variance Table

| Source of Variation | DF | MS | F | P |
|---|---|---|---|---|
| Sociability main effects | 2 | 0.10 | 0.03 | 0.97 |
| Treatment main effects | 2 | 0.62 | 0.18 | 0.83 |
| Sociability X Treatment interaction | 4 | 3.03 | 0.90 | 0.47 |
| Error | 52 | 3.38 | | |

n = 61

Table 14. Summary two way analysis of variance table for academic achievement posttest X #1 in relation to level of sociability and treatment group.

Summary Analysis of Variance Table

| Source of Variation | DF | MS | F | P |
|---|---|---|---|---|
| Sociability main effects | 2 | 11.68 | 3.22 | 0.04* |
| Treatment main effects | 2 | 1.30 | 0.36 | 0.70 |
| Sociability X Treatment interaction | 4 | 1.97 | 0.54 | 0.71 |
| Error | 52 | 3.63 | | |

n = 61

* significant at $p<0.05$

# COMPUTER MANAGED LEARNING AS FACILITATOR OF INDIVIDUALIZED LEARNING IN TEXTILE SCIENCE

E.M. Crown and Maryanne Doherty
Faculty of Home Economics, The University of Alberta

Edmonton, Alberta, Canada T6G 2N1

ABSTRACT

The purpose of this research was to evaluate a specific computer managed learning (CML) system for its effectiveness in facilitating individualized learning for students in an introductory textile science course. The evaluation design incorporated three sets of variables: antecedents, transactions and outcomes. Data for the evaluation were obtained from students, resource persons and the course professor. Most of the standards set in the evaluation design were achieved; thus both the modular approach to instruction and the CML system were judged to have been successful for the course.

This paper describes one portion of the evaluation research that was conducted as part of a project entitled "The Development of Demonstration Learning Systems for Home Economics Programs"[1]. The project focused on courses that had high enrollments and/or students with varying academic and experiential backgrounds and varying career aspirations. Because the project leaders wanted to better meet the individual needs of the students, they sought to develop innovative teaching/learning systems.

## Learning Systems

According to Kozma, Belle and Williams (1978:12) the complex, unique and interdependent nature of the instructional situation can be represented by a system. Kozma included six elements in a learning system: the instructor, the learner, the subject matter content, the medium, evaluation, and the social and physical environment.

The teaching/learning systems included in the project were characterized by a modular approach, which according to Cross (1976) is a system of self-contained learning modules with well-defined objectives, usually consisting of learning materials, a sequence of learning activities and provisions for evaluation. Cross and others (Donald, 1973; Goldschmid & Goldschmid, 1972) have described the modular approach as a type of learning system that promotes individualized learning. The learning system for the introductory textile science course described in this paper consisted of thirteen learning modules.

---

[1] The project was carried out in the Faculty of Home Economics, The University of Alberta, 1979-1981, and was funded by an Innovative Projects Grant from Alberta Advanced Education and Manpower.

CML as Facilitator. In the textile science part of the project, computer managed learning (CML) played an important role in the teaching/learning system. When instruction is individualized for a large number of students (enrollments for this course have varied from 70 to 110 students) some management system is needed. CML allows for monitoring of student progress by the instructor, and allows individual students to test competency, monitor their own progress, and manage learning experiences according to individual competencies and needs.

The specific system used in this project was the PLATO Learning Management (PLM) system. Five terminals were available in the textile laboratory, and other terminals were accessible to students elsewhere on campus. Students selected one of three management strategies with varied learning objectives, depending on their backgrounds and intention to study further in textile science. Students could take a pretest before attempting any module and could take self-assessment tests for each module until all objectives had been mastered. For unmastered objectives, additional learning activities were recommended. In addition, three credit tests and a final examination were scheduled throughout the term. All such tests consisted of items selected from the computerized item bank and a small number of instructor-marked questions.

## Evaluation Design

In order to determine the effectiveness of the learning system, an evaluation design was developed based on the Stake (1967) model as a conceptual framework. The model emphasized two essential operations in the evaluation process: description and judgement. Both operations considered three sets of variables: antecedents, transactions and outcomes. The variables considered are listed in Table I. In the judgement process, the observed data were compared to a set of standards made explicit for each of the variables as part of the evaluation design. For example, for variables which were measured by student ratings, the standards were set at 80% agreement by students.

Evaluation instruments included a student information form, an attitude scale and a competency self-rating scale completed by students at the beginning of the course, module evaluations completed by students throughout the course, and a professor questionnaire, a resource person questionnaire and a student questionnaire completed at the end of the course. The attitude scale and competency rating scale were completed again by students at the end of the course.

Although the complete evaluation design was outlined above, only data relating to those variables most relevant to the focus of this paper - CML as facilitator - will be reported below.

TABLE 1. VARIABLES SELECTED FOR EVALUATION DESIGN

| | |
|---|---|
| Antecedents | Student Population Entry Characteristics<br>- Academic Background<br>- Faculty of Registration<br>- Major Area of Study<br>- Year of Program<br>- Attitude Toward Textile Science<br>- Competency (self-rating) in Textile Science |
| Transactions | Student Involvement with Modularized Learning<br>Student Involvement with CML<br>Student Involvement with Other Students<br>Student Involvement with Professor and Resource Person |
| Outcomes | Student Ratings of Each Module<br>Effectiveness of Modules in Meeting Stated Objectives<br>Attitude Toward Textile Science<br>Competency in Textile Science |

## Results and Discussion

The observations for each of the relevant variables are summarized in Table II. More details regarding each of the observations are reported in Doherty (1981).

The course professor, one resource person and a large majority of students rated the use of modules as effective, efficient and appropriate. One resource person was less sure of the appropriateness of modular learning. This same resource person was unsure of the effectiveness and appropriateness of CML and strongly disagreed that CML was efficient. Since the PLATO system was new to the University at the time of the project, the system was down periodically. The resource person who was least satisfied had experienced the most problems with such system breakdown.

Such problems do not appear to have caused much dissatisfaction among students. Student ratings of CML are summarized in Table III. A large majority of those students who responded agreed that CML allowed efficient access to self-testing, self-paced learning and self-tracking, and was appropriate and enjoyable. Many of the students elected to take credit tests and the final examination before the scheduled dates for such tests.

The professor was quite satisfied with CML despite the many extra hours required of her to implement the system, and despite the fact that PLATO did not provide a completely satisfactory record-keeping system, in that there was no provision for incorporating grades from instructor-marked assignments. Such a program has since been developed for PLM at this University, providing a much more efficient management system.

The 80% standard was not met regarding student interactions with other students, resource persons or the instructor, although a majority of students agreed that their interactions were satisfactory (Table III). One benefit claimed for CML is that it should free the instructor, providing more time for interaction with individual students. It was noted in this course, however, that students were unused to such a system, and very few of them took advantage of the time available for individual consultation. It is possible that they did not find such consultation necessary.

A large majority of students believed that the modules were effective in helping them to meet the stated objectives for each module and in increasing their competencies according to their own needs. In fact, although not recorded as part of the original evaluation design, of the 84 students registered in the course only one did not meet the pass criterion, and 57% of the students obtained over 75% on the course.

At the end of the course students indicated a more positive attitude toward textile science than at the beginning, for each of the 22 items in the attitude scale. For 11 of these items, the change was statistically significant ($p < .05$). When students were grouped according to major area of study, each group indicated a more positive attitude for each of the 22 items.

At the end of the course, students rated themselves much higher on the 14 competencies than they had done prior to taking the course. Although the ratings almost doubled (on a scale of 1 to 5) for each of the competencies, these increases were statistically significant ($p < .05$) for only three of the competencies. Again, when students were grouped according to major area of study, students in each group rated themselves much higher on each competency at the end of the course than they had done prior to taking the course.

## Conclusions and Recommendations

Most of the standards set in the evaluation design were achieved; for many variables the results were much better than the 80% standard. Thus, the modular approach is considered to have been successful for this course, and the CML system is judged to have played a major role in facilitating the effective implementation of individualized instruction.

The variables for which the standard was not achieved were student interactions with other students, the resource person and the professor. In future, more effort should be made to clarify for the students the role of the resource person and professor.

It is recommended that the modular system, the related learning materials and CML continue to be used for the introductory textile science course, and that evaluation be ongoing to supply feedback for further revisions.

## References Cited

Cross, Patricia. Accent on learning. San Francisco: Jossey-Bass, 1976.

Doherty, M. An evaluation of modularized systems. Unpublished doctoral dissertation, University of Alberta, 1981.

Donald, J.G. Modular instruction: a resource book. Montreal: Center for Learning and Development, McGill University, 1977.

Goldschmid, B. and Goldschmid, M.L. Modular instruction in higher education: a review. Montreal: Center for Learning and Development, McGill University, 1972.

Kozma, R.B., Belle, L.W., and Williams, G.W. Instructional techniques in higher education. Englewood Cliffs, N.J.: Educational Technology Publications, 1978.

Stake, R. The countenance of educational evaluation. Teachers College Record, 1967, 68: 523-540.

TABLE II. OBSERVATIONS: TRANSACTIONS AND OUTCOMES

| | VARIABLES | OBSERVATIONS |
|---|---|---|
| TRANSACTIONS | Student involvement with modularized method of teaching/learning. | 94% of students agreed that the use of modules was effective; 87% agreed that the use of modules was efficient; and 87% agreed that the use of modules was appropriate.<br>Both resource persons agreed that the use of modules in this course was effective and efficient. Only one agreed that the use of modules was appropriate.<br>The professor strongly agreed that the use of modules in this course was effective, efficient, and appropriate. |
| | Student involvement with CML system. | 83% of students agreed that the use of CML allowed efficient access to self-testing; 86% and 89% respectively agreed that the use of CML allowed self-pacing and self-tracking; 89% agreed that the use of CML was appropriate; and 86% agreed that CML was an enjoyable experience.<br>One resource person agreed that the use of CML in this course was effective, efficient and appropriate; the other resource person was uncertain about the effectiveness and appropriateness of CML and strongly disagreed with the efficiency of CML in this course.<br>The professor strongly agreed that the use of CML was effective, efficient, and appropriate. |
| | Student interactions with other students. | 75% of students agreed that their interactions with other students were satisfactory, 21% were uncertain, and 4% disagreed. |
| | Student interaction with resource persons. | 70% of students agreed their interactions with the resource person was satisfactory, 15% were uncertain and 15% disagreed.<br>One resource person agreed that interaction with students was satisfactory and one disagreed. |
| | Student interaction with the professor. | 54% of the students agreed that their interactions with the professor were satisfactory, 17% were uncertain and 28% disagreed.<br>The professor strongly agreed that her interactions with the students in her laboratory sections (about half the students) were satisfactory, but could not rate her interactions with the other students. |

TABLE II (Continued)

| | VARIABLES | OBSERVATIONS |
|---|---|---|
| OUTCOMES | Effectiveness of modules in meeting stated objectives. | From 83% to 96% of students agreed that they had met the stated objectives for each of modules 1 to 12; 73% agreed for module 13 while 27% were uncertain. |
| | Effectiveness of modules in increasing student competency. | From 84% to 100% of students agreed that each of modules 1 to 12 were helpful; 50% agreed for module 13, while 32% disagreed.<br>94% of students agreed that the use of modules in the course allowed them to increase their competencies according to their own needs. |
| | Student population exit characteristics - attitude toward textile science. | Students did not indicate a significantly more negative attitude toward textile science on any of the attitude statements or pairs in the attitude scale, compared to the pre-course measure. Post-attitude measure was significantly ($p < .05$) more positive for 6 of the 14 attitude statements and 5 of the 8 attitude pairs. |
| | Student population exit characteristics - competency in textile science. | Students did not rate themselves significantly less competent on any of the 13 competencies compared to pre-course ratings, but rather rated themselves significantly ($p < .05$) higher on 3 of the competencies.<br>Only one student did not meet the pass criterion; 57% of students obtained over 75% on the course. |

TABLE III. STUDENT RATINGS OF CML

| | Responses | | | | |
|---|---|---|---|---|---|
| | Strongly Agree | Agree | Uncertain | Disagree | Strongly Disagree |
| CML allowed efficient access to self-testing | 36% | 47% | 4% | 13% | 0 |
| CML allowed self-paced learning | 43% | 43% | 10% | 4% | 0 |
| CML allowed self-tracking | 47% | 42% | 6% | 6% | 0 |
| CML was appropriate | 40% | 49% | 4% | 8% | 0 |
| CML was enjoyable | 43% | 43% | 8% | 2% | 4% |

# NATIONAL SURVEY FOR COMPUTER-BASED EDUCATIONAL SOFTWARE IN HOME ECONOMICS

Frances Whitaker Mayhew

University of Delaware

ABSTRACT

National Survey for Computer-Based Educational Software in Home Economics. A survey was made of all college-level academic units of Home Economics and Cooperative Extension Services listed with the American Home Economics Association to locate the utilization and/or development of Computer-Based Education with content related to Home Economics. The response revealed CBE lessons in all but two of the AHEA designated subject matter areas. The Cooperative Extension Services were shown to be extremely active in CBE utilization. A high degree of computer illiteracy was evident and much duplication of effort, especially in Food Science and Consumer Economics, was revealed.

The utilization of the computer for instructional purposes seems to be gaining popularity each year. Surely as the computer itself becomes more and more instrumental in the routine management of our daily lives, in the business sector, and in the general educational scheme of today's children, it will become a standard component in the management of personal information in the home. Since home economists identify with factors which impact on the quality of life, this growing fascination with the computer and its capabilities are naturally bound to interest us.

As educators we see the potential in the computer as a teaching tool. Many innovators among us have already begun to develop software for use in their classes. Others may have desired to do so, but found the resources of time and technical assistance unattainable. The development of computer-based educational software is time consuming and may be unnecessary if suitable programs can be found. The benefits of objective peer evaluation and elimination of duplication can accrue through utilization of existing software. The major obstacle to this procedure is the absence of a convenient means to obtain information relative to software development.

This report presents the current findings of a project which may well be continued for many years. It is an attempt to locate and catalog computer-based instructional software which has been developed, or is presently being used, to teach home economics related concepts to the elementary, secondary, college and adult audiences. The many benefits of computer-based instruction are documented in scholarly literature, but the development of sound software is a time consuming venture. The location of existing CBE material is equally time consuming since standard resources which are indexed within computer search databases reveal few items listed related to learning experiences in the field of home economics which involve a computer. Either these innovators who are developing home economics related C-B-E are:

- Not willing to share their experiences with others
- Too busy with other responsibilities to prepare articles for publication or,
- Have not been able to locate a scholarly journal which will accept their article.

For whatever reason, the development of CBE in home economics areas is not being publicized, thereby allowing a considerable duplication of effort in specific content areas to occur (notably food science and consumer economics).

The pursuit of this survey fulfills two goals of the SIG/HomEc:

1. The desire of the Home Economics Special Interest Group to act as a clearing house for information related to CBE in home economics, and;
2. The hopes of the SIG/HomEc to forestall duplication of effort through interactive communication among developers.

Another factor motivating the compilation of existing software into a catalog is the observation that computer searches reveal minimal documentation of C-B-E development in home economics

The instrument used in this survey was a simple three item questionnaire. This questionnaire was accompanied by a cover letter explaining the objective of the survey and the affiliation of the researcher to ADCIS. An ADCIS brochure accompanied each mailing, so that the survey would convey the affiliation with the organization.

The three questions asked were these:

1. Was any computer-based educational material being developed within this unit, and if so, by whom?
2. Did the unit own a micro-computer, if so, what system?
3. Would the respondent desire a copy of the compiled listing of C-B-E software once it was completed?

If the respondent indicated that any computer-based-educational materials were being developed within his unit, he was asked to complete an additional form providing information related to this lesson such as:

-- lesson title, objective and level of difficulty
-- technical information such as system, language, peripherals, etc.; and
-- person to contact for further information.

The population for this survey was taken from a listing provided by the American Home Economics Association. This list consisted of administrators of academic units of home economics at the college level and state leaders of cooperative extension services throughout the country. This listing included 116 college level units of home economics instruction and 35 units of cooperative extension service. Using this list 151 mailings were sent. One hundred and sixteen academic units and 37 cooperative extension units were contacted. This listing did not provide addresses for academic units in one state (Alaska) and Cooperative extension units in 14 states. It is possible that some of those missing cooperative extension units may be administered by the Dean of the land grant college, but none of these cooperative extension units have responded to the survey questionnaire.

A 66% response was observed at the time of this writing (2/82), with 76 institutions returning the questionnaire. Fifty-three of 116 or 45% of the academic units and 23 of 35 or 65% of the Cooperative Extension Service units have responded. Of these respondents 44 indicated that they were developing computerized instructional materials, providing records of 137 individual lessons. This number indicates the individual lessons reported and catalogued at this writing. Many lessons were reported in groups or as networks with only fragmented information regarding the individual lessons. After follow-up contacts the complete information on these additional lessons will also be recorded. The catalog listings will also be enlarged by additional lessons obtained from other sources such as overlapping content-area groups (Health Educational Network) and software manufacturers.

The lessons are currently classified as to content by the AHEA Subject Matter Area designation. The categorical listing of lessons currently catalogued is as follows:

CATEGORY 1 Art (2) (total)
-- Drafting

Category 2 Family Economics and Home Home Management (26)
-- Cash Flow Summary, Budgetary Analysis and Guidelines
-- Spending Patterns
-- Mortgage Level Based on Income

2. FAMILY ECONOMICS AND HOME MANAGEMENT (26)
-- Cash Flow Summary, Budgetary Analysis and Guidelines
-- Spending Patterns
-- Mortgage Level Based on Income
-- Cost/Benefits of Home Ownership
-- Cost of Credit
-- Cost of Child Raising
-- Food Budget Considerations
-- Family Resource Allocation (Time, money, etc.)
-- Inflation (CPI)
-- Consumer Behavior

3. FAMILY RELATIONS AND CHILD DEVELOPMENT (13)

    -- Developmental Profiles of Handicapped Children
    -- Child Development (2-5 years)
    -- New Baby/Mother Relationships
    -- Simulated Observation Techniques Training
    -- Inventory of Family Events
    -- Resources for Stress Management
        -- Duel Employment
        -- Chronically Ill Child

4. FOOD AND NUTRITION (50)

    -- Food Preservation (ingredient manipulation)
    -- Functional Properties of Foods
    -- Food Management
    -- Dietary Intake Analysis
    -- Quantity Foods
    -- Unspecified Objective
    -- Energy Balance
    -- Nutrition Elements
    -- Fluid Chemical Balance
    -- Physiological Chemistry of Nutrition
    -- Metabolism
    -- Pediatric Nutrition
    -- Menu Evaluation
    -- Recipe Nutrient Evaluation
    -- Effect of Cooking Method on Vegetables
    -- Evaluation of Nutrient Content and Cost

5. COMMUNICATIONS - 0

6. TEACHER EDUCATION (5)

    -- Decision Making
    -- Multiple Choice Test Performance
    -- Test Item Development
    -- Objective Development
    -- Problem Solving Approach

7. HOUSING, FURNISHING AND EQUIPMENT (11)

    -- Cost of Home Ownership
    -- Home Energy Efficiency
    -- Renting vs Owning Housing
    -- Kitchen Design
    -- Basic Household Electricity

8. INSTITUTIONAL MANAGEMENT (3)

    -- Forecasting Dollar Values and Menu Item Sales
    -- Food Service Accounting

9. TEXTILES AND CLOTHING (11)

    -- Basic Textiles
    -- Stain Removal
    -- Business Mathematics
    -- Merchandising
    -- Pattern Drafting
    -- Pattern Selection
    -- Comfort
    -- Costume History

From this superficial data analysis it can be seen that a fair amount of developmental activity is occurring in the area of computer-based educational materials with home economics content. As the search continues, it seems clear that considerably more software will be found. The elementary and secondary level teachers, whose strength is their creativity, and the majority of the software manufacturers have yet to be surveyed. This work has to be considered as forever ongoing to continually seek-out and enlarge the body of available computer-based educational materials available to educators in home economics.

The results of this survey show the concentration of effort within certain specific content areas which might have been eliminated if the authors had been aware of the efforts of each other. This duplication of effort will hopefully be reduced once the catalog of lessons becomes available. Then authors with similar ideas can contact one another and draw from the experiences of each other rather than duplicate the same errors.

At this point in the development of this survey, several concerns have arisen:

-- first, there must be some way to contact computer-based education developers at the elementary and secondary level,

-- second, we as home economists must find a vehicle through which we can communicate our successes and failures with C-B-E. The newly established SIG/Hom/Ec Newsletter is one receptive publication for comment and discussion. The Journal of Computer-Based Instruction is another. The Journal of American Home Economics Association is a third possible vehicle.

-- Third, once appropriate software has been located, the next concern is to the quality of this material. Much computer-based material is published after only a technical review, with no systematic evaluation of the educational soundness of the content material. We, as Home Economists should concern ourselves with the quality-of-content of this educational material. We have the content expertise that most software manufacturers do not.

-- The fourth concern is toward computer literacy.

Of the lessons reported, 85 were

designed for mainframe systems, 27 were designed for micro/mini systems. Twenty-five of the lessons reported were linked to inappropriate or unclassifiable systems. This indicated a glaring lack of computer literacy which is not unexpected among authors whose strength lies in content and educational skill. Most authors of this type, especially within a university setting, will seek out the technical assistance of computer center personnel when developing a lesson, thereby acquiring little knowledge of the computers capabilities which could be utilized for teaching.

None the less, the burgeoning interest and productivity among home economists and others in related fields is clearly shown. Many individuals have already contributed to this catalog of lessons and as the search continues, the catalog will grow. For further information related to this catalog please contact:

Frances W. Mayhew
Alison Hall
University of Delaware

# SIMULATED LABORATORY EXPERIMENTS INVESTIGATING FUNCTIONAL PROPERTIES OF FOODS

Zoe Ann Holmes

Oregon State University

## ABSTRACT

A computer package consisting of simulations and associated instructional materials was developed to provide the student with the opportunity to learn basic information and gain expertise towards solving problems which occur in foods. The flexible computer model includes 26 simulations covering a range of food items such as sugars, flour mixtures, eggs,milk, meat,fish, fruits and vegetable products. The simulations give data related to the effect of selected treatments on properties, characteristics and/or sensory parameters being tested. Evidence indicates that the simulations can be a successful method of teaching. Inclusion of the simulations has been shown to significantly improve the students attitude toward the course at Oregon State University titled "Science of Foods."

## INTRODUCTION

The use of computers to develop and implement predictive models or simulations for the nutritional and esthetic quality of foods has increased rapidly in both industrial and education settings. Workers have used the computer to model protein interactions in angelfood cakes (Johnson and Zabik,1981). Other workers have used computer modelling or simulations to formulate whipped toppings (Mir and Thomas, 1980),develop a sausage-type product from minced mullet (Daley et al., 1978) and produce whey protein concentrate (Unknown, 1982). A recent survey of computers in undergraduate food science education has emphasized that approximately 75% of the B.S. graduates in food science receive some form of computer education (Steff, 1981). The development of this project which was concerned with simulated laboratory experiments in food science arose out of an interest in this aspect of computer use and a need to solve educational and budget problems in the "Science of Foods" course. The objective was to structure the course so that the student would: familiarize themselves with a variety of physical, chemical, biochemical and/or physicochemical properties of foods as influenced by a variety of conditions; solve problems based on knowledge of factors influencing the quality of foods; gain experience in the design of a variety of experiments; become familiar with an interactive computer system. It appeared that the development of a computer model focusing on simulated laboratory experiences would be appropriate.

## COMPUTER PACKAGE

The computer package consists of a Student Guide, Instructor Guide and the model. Each of these components were developed from analysis of the available literature. The computer model consists of twenty-six separate simulations (Table 1). Each simulation covers a distinct area such as sugars, eggs, meats and vegetables.

Each of these simulations (Table 1) consists of a range of values representing data for characteristics, properties of foods. The CYBER CDC 720 computer randomly selects values from the range of data when a student does a simulated laboratory experiment. Each replication for each treatment yields a different data point. Any experimental or literature data could be used to create such realistic simulated laboratory experiments. Each of the twenty-six simulations using the CYBER computer is interfaced with the Michigan Experimental Simulations Systems to furnish the requisite programming. Since the development of the original programming in 1979, the concept and data values have been adapted to a Radio Shack Microcomputer with success.

TABLE 1. SIMULATIONS AVAILABLE IN THIS MODEL.

EGGS-MERINGUES
EGGS-ANGELFOOD CAKE
EGGS-CUSTARD
MEAT-GENERAL
MEAT-BEEF
MEAT-PORK
MEAT-FISH, LAMB, MUTTON
MEAT-POULTRY
FRUITS-GENERAL
FRUITS-PROCESSED
VEGETABLES-GENERAL
VEGETABLES-PROCESSED
EGGS AND DAIRY PRODUCTS-COMPOSITION,PROPERTIES
BRAIN
SUGAR-PROPERTIES
SUGAR-CRYSTALLIZATION
WATER-CRYSTALLIZATION
FATS AND SHORTENINGS
STARCH
FLOUR MIXTURES-GENERAL
FLOUR MIXTURES-PASTRY
FLOUR MIXTURES-BREAD
FLOUR MIXTURES-BISCUITS
FLOUR MIXTURES-MUFFINS
FLOUR MIXTURES-SHORTENED CAKE
DAIRY PRODUCTS

In addition to the computer model, the two document sources are critical to the success of the simulations. The Student Guide contains instructions for doing the simulations, a list and description of treatments, tests, properties, recipes, and quality characteristics. Additionally, a review of basic knowledge required for each simulation is included. The Instructor Guide includes activities, the numerical data range and an annotated bibliography for use with each simulation.

## SIMULATION USAGE

In using the simulations, students are given a problem or activity selected to accomplish a specific goal or course objective. For example, in using the simulation FLOUR MIXTURES - SHORTENED CAKE, a range of activities varying in complexity (Figure 1) could be done. For instance, Example A (Figure 1) might be used in a beginning foods class and Example B in an experimental foods class. To solve either problem, the student would generally follow the flow diagram shown in Figure 2 as the decision-making process. The student would make decisions based on information in the Student Guide (Figure 3), from the instructor and outside reading. Learning for the student comes from the student solving the problem presented, obtaining the data and its subsequent analysis, evaluation and presentation.

## SIMULATION EVALUATION

These simulations were evaluated in the "Science of Foods" course at Oregon State University. Each simulation cost approximately $0.75 (7.5 SRU) to use. The simulations were used in association with the lecture method and guided design (Wales, 1976) when

EXAMPLE A.

Hillary was a consultant for President Reagan. The Reagans' liked to have cake on their menu; however, because of the publicity they decided they had better show their commitment to "economy" by cutting down from a rich ratio cake to a leaner formula. Before making this drastic change, President Reagan decided to show his concern and hired Hillary to run a series of tests to evaluate the change. Taking the role of Hillary, do the following:

°Using the computer simulation FLOUR MIXTURES-SHORTENED CAKE of the Student Guide to Simulated Laboratory Experiments Investigating the Functional Properties of Foods, plan the treatments and factors to do the requisite experiment.

°Run the Simulation.

°Write-up the RESULTS in a professional manner in the following two formats.
1. Write the INTRODUCTION, EXPERIMENTAL, RESULTS AND DISCUSSION, REFERENCES using the IFT Style Guide for the Journal of Food Science.
2. Write the "press release" for President Reagan to release.

Figure 1. Examples of activities used by simulation. (Continued)

(Figure 1. Continued)
EXAMPLE B.

Edward understood that as a batter was more viscous, its specific gravity would change which would result in changes in the tenderness and fluffiness of the cake. Develop an experiment which would evaluate the interrelationship(s) between batter specific gravity, batter viscosity, cake tenderness and cake lightness.

- °What treatment(s) should Edward choose?
- °What factor(s) should Edward choose for investigation?
- °How many replications should Edward choose?
- °Run the experiment on the computer simulation.
- °Discuss these interrelationships, indicating why the interrelationship(s) are valid.

Figure 1. Examples of activities used by simulation.

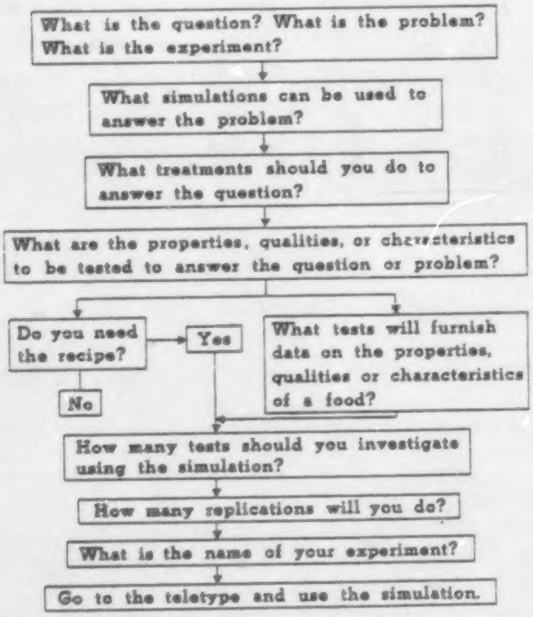

What is the question? What is the problem? What is the experiment?
What simulations can be used to answer the problem?
What treatments should you do to answer the question?
What are the properties, qualities, or characteristics to be tested to answer the question or problem?
Do you need the recipe?
Yes
No
What tests will furnish data on the properties, qualities or characteristics of a food?
How many tests should you investigate using the simulation?
How many replications will you do?
What is the name of your experiment?
Go to the teletype and use the simulation.

Figure 2. Flow diagram of steps required before initiation of computer use phase of simulation.

appropriate for completion of an objective. Of the 44 undergraduate students, 91% were shown to be capable of handling the abstract-formal mode of instruction utilized in the course. This level of thinking was determined through the use of a Paiget Test. Pre- and post-attitude and achievement tests were given to the students at the beginning and end of the course. Of the 27 attitude questions, the 9 questions shown in Table 2 indicated a significant change towards strongly agree in course attitude. A key to this positive change in attitude was partly dependent on initial instructor assistance during the first simulation. Of the 25 pre- and post-achievement test questions, 18 questions, 17 represented by computer simulations, showed an improvement in knowledge. The other achievement questions revealed no significant difference as students had initially identified the correct answer.

## REFERENCES

Daley, L. H., Deng, J. C. and Cornell, J. A. Development of a sausage-type product from minced mullet using response surface methodology. Journal of Food Science, 1978, 43; 1501.

Johnson, T. M. and Zabik, M.E. Response surface methodology for analysis of protein interactions in angel food cakes. Journal of Food Science, 1981, 46; 1227.

Min, D. B., and Thomas, E. L. Application of response surface analysis in the formulation of whipped topping. Journal of Food Science, 1980, 45; 346.

Steffe, J. F. Computers in undergraduate food science education - a survey. Food Technology, 1981, 35; 98.

Unknown. Special membrane cartridge provides efficient automated whey processing. Food Engineering, 1982, 54(1); 154.

Unknown. Jean Piaget's test of concrete and formal reasoning pattern. Reasoning Workshop. Berkeley: University of California, 1975.

Wales, C.E. Guided Design. Morgantown, West Virginia: West Virginia University, 1976.

Figure 3. Simulation FLOUR MIXTURES-SHORTENED CAKE, treatments, and properties,tests or characteristics available in the computer model.

| TREATMENTS | PROPERTIES, TESTS, CHARACTERISTICS |
|---|---|
| standard shortened cake | recipe |
| lean ratio shortened cake | pH, batter |
| rich ratio shortened cake | specific gravity |
| cake with unbleached flour substituted | viscosity, batter |
| cake with hard wheat flour substituted | internal temperature |
| cake with high fructose sugar substituted | compressibility |
| cake with 6 percent emulsifier added | compressibility, area under curve |
| | volume |
| | pH, crumb |
| | C.I.E. crumb color, X value |
| | C.I.E. crumb color, Y value |
| | C.I.E. crust color, X value |
| | C.I.E. crust color, Y value |
| | sensory evaluation, crumb color |
| | sensory evaluation, cell compactness |
| | sensory evaluation, cell uniformity |
| | sensory evaluation, flavor |
| | sensory evaluation, moistness |
| | sensory evaluation, tenderness |
| | sensory evaluation, shape |

Table 2. Students' responses, pre- and post-test, to attitude statements related to value of objectives and computer simulation activities

| question | mean score*† pre-test | mean score*† post-test |
|---|---|---|
| I (will find, found)# it easy to study this subject because I (am, was) really enthusiastic about it. | 1.9 | 1.5 |
| I (think, thought) I (learn,learned) a lot in this class. | 1.9 | 1.0 |
| This course (should give, gave) me a good perspective of foods information required of dietetics or foods and nutrition majors. | 1.5 | 1.0 |
| The subject matter of this course (is, was) interesting. | 1.6 | 1.0 |
| The class (seems, seemed) to be well organized. | 2.0 | 1.6 |
| The course (should present, presented) an adequate survey of the subject matter areas important in food preparation. | 1.5 | 1.2 |
| I am aware of the types of competence in foods which will be expected of me when I graduate with a Bachelor of Science degree. | 2.0 | 1.7 |
| I feel confident that this course enabled me to gain valuable skills and competencies that will be of benefit to me in the field of dietetics or foods. | 1.5 | 1.2 |
| I (will be, am) able to get help from the instructors whenever I needed it. | 1.5 | 1.2 |

*N=44; Significant difference ($p \leq 0.05$)
†Response: 1=strongly agree; 2=agree; 3=neutral; 4=disagree; 5=strongly disagree.
#(Pre-test verb, post-test verb).

# STUDENT EDITED GRAPHICS

Thomas J. Chapin and Richard W. Davis

Bell System Center for Technical Education

## ABSTRACT

Greater student learning and performance can be achieved through the use of computer-based simulation, and greater simulation can be produced by allowing students to use simple graphics editors to create and manipulate graphics displays, and allowing the computer to judge the results.

To simplify the process of creating graphics editors for students, a conceptual schema is provided to analyze the objects to be manipulated, the manipulations available, and potential judgements of the results of student manipulation.

## INTRODUCTION

During the past few decades, instructional developers and designers have come to realize the importance of graphic, nonverbal information in student learning. We can summarize a huge amount of research and writing by a paraphrase of an old maxim:

> One picture is worth a thousand words, and an animated graphic may be worth a thousand pictures.

Our own work with Computer Based Instruction at the Bell System Center for Technical Education (BSCTE) in Lisle, Illinois, has involved us very heavily in the use of graphics. Our work spans many systems and approaches, but the particular projects we wish to discuss here are recent efforts using the PLATO system to teach technical skills to Bell System managers.

## STUDENT PERFORMANCE

The goal of all of our instruction at BSCTE is to improve job performance. While the formal methods we employ in designing courses fill fourteen inches of reference binders, the basic approach is as simple as this:

- Determine what the student's job is.
- Determine what skills and knowledge are needed for the student to perform that job successfully.
- Design ways to provide skills and knowledge to the student in useful and effective ways.
- Provide opportunities for the student to practice the performance of those skills, and ways to test the student's ability to perform.

Under this scheme of instruction, the student performs, in a simulated environment, the same activities to be performed on the job. Computers have allowed us to achieve higher levels of simulation than available through other means of training development and delivery.

## SIMULATIONS

In our initial efforts with PLATO, we used computer graphics to represent paper graphics which the student would encounter on the job. We were simulating only one small aspect of the job environment--the technical documentation--especially on tests.

But even this fledgling effort involved real simulations and proved effective. If students worked on the job with a circuit diagram or a chart, we wanted to simulate that document very accurately; we wanted to provide an easy transition for the student between what they did in the computer lesson, and what they did on the job.

And we found that, in fact, we could do a very good job of writing lessons with high "training transfer" by taking advantage of the graphics display on the PLATO IST II terminal. The computer graphics were good enough that we often included print reproductions of the terminal display in our written lesson materials. (See figure 1).

As our sophistication in lesson writing grew, we worked more and more intimately with the interaction or communication between our students and the graphics displays we were using to teach them. We began to explore simulating those aspects of jobs in which students not only obtained information from graphics, but created them or manipulated them as well.

So we became concerned with students using graphics, not merely recognizing and understanding graphics presented to them. As a result of these efforts, we found ourselves able to use graphics manipulation to teach students more than "signal learning" (Gagné, 1970), and were able to provide a higher level of job simulation than previously.

## STUDENT GRAPHICS EDITORS

While reflecting upon our activities, we asked ourselves this key question: How do we want the student to respond to a graphic display? The answer became clear: We want them to respond graphically.

Specifically, we want students to be able to recognize the elements in a graphics display, manipulate those elements, draw elements into relationships, and create new elements. In doing so, the students would, in reality, be manipulating the information represented by the elements of the graphic display.

But could we get the computer to support students' activity in creating or manipulating graphics? The answer was clearly YES, because in fact we were doing exactly that ourselves as authors whenever we created or edited a computer graphics display. Furthermore, we had seen some of the capabilities we wanted in at least rudimentary form in such well known PLATO lessons as "Titrate," "Fruitfly" and "Sentences." Our challenge was to develop ways to do two specific things the computer had not been asked to do before:

- Let students use simple graphics editors to manipulate displays.
- Make the computer judge the graphic display to determine if the desired performance occurred.

This was a significant undertaking.

### Actions of the Editor

Today we have several courses using these graphics techniques. We have tools for developing the courses and a conceptual rationale to help us think about what we are doing.

### Objects

The objects on the display can be conveniently grouped into three categories:

- icons - conventional pictorial symbols or representations
- alphanumerics - verbal symbols included within graphics
- vectors - lines with indicated or implied meaning

### Object Manipulation

The student can manipulate these elements in six ways:

- Locate (or select) an item
- Create an item
- Delete an item
- Relocate an item
- Change information
- Relate items to each other using vectors

Several of these manipulations have analogs in conventional test item construction. Selection, for example, is the graphics equivalent of the multiple choice question. But the use of icons or vectors in a display permits higher levels of simulation and higher levels of learning on a hierarchy. In addition, selection of graphics items may appear more friendly towards the student than text-only versions of test items.

For instance, we frequently deal with circuit analysis. Instead of giving the student lists of possible locations where some action is to be taken in the circuit, (a,b,c, etc), we show the student the circuit drawing and ask him or her to touch the proper location. Some special graphic effect emphasizing the student's choice will both gain interest and reduce accidental errors. The direct graphics response may eliminate many intermediate mental steps for the student needed solely to satisfy the limitations of the computer's input capability.

## Creation

Very high degrees of simulation are achieved if the student is allowed to create the graphic itself or to manipulate it realistically so that the elements are in new and "meaningful" relationships.

The student may place an icon, text string, or vector in the proper place on the display, or may move an element from an incorrect location to a correct one.

And when objects are in place, the student may then, through whatever means are given, draw the objects into the proper relationships with each other using vectors or relative positioning.

## JUDGING

One can easily imagine how complex the judging of graphics could be. To actually give the computer "understanding" of the graphic--to recognize objects and their relationships --would challenge even the most sophisticated artificial intelligence systems. But our move from judging verbal or string messages to judging graphic messages was quite easy. The reason is that we created a solid conceptual rationale for our thinking about how to judge displays.

Potential student errors are of three types:

- Wrong information given
- Missing or extra information
- Improper relationships indicated.

First, the icon, text, or vector selected (or created) by the student may be incorrect. That is, it does not match that given in the answer key. If the selection is among given icons or vectors or among a limited selection of display locations, the judging process can be done easily by the computer.

Second, an icon, text, or vector may not have been supplied when needed or may have been supplied when unwanted. Or in relative terms, an element may be be extra or missing in relation to another element. Again, judging the finite number of icons in the menu supplied to the relatively small number of possible display locations is easy for the machine.

Finally, elements of the display may have been drawn into relationships with one another improperly. If one thinks of a vector as a special icon which has two locations on the display (its endpoints), one can see that it could be judged using the same schemes as employed for icons.

## APPLICATIONS

At BSCTE, in 18 months we have progressed through three phases of student manipulation of graphics, from simple selection, through creation and location of elements in a display, to creation of entire displays by the student with all elements in indicated relationships.

At first, we used simple variations of multiple choice questions where, instead of choosing among text items, the student was asked to touch or otherwise indicate their answer as a section of a circuit diagram which lit up to indicate their choice. (See figure 1)

In the middle phase, we began to give students the ability to place objects, text and icons, on the display. The student was given a schematic workprint similar to that used on the job, and was asked to complete it by adding symbols and text wherever appropriate. (See figure 2)

Now, we have allowed the student to create entire displays on the screen, and to move elements around at will to indicate desired relationships. The student is asked to create a flowchart such as they would on the job. They use a given set of icons which they may place on the display as they wish (within certain limitations). Their work is judged on the basis of the logical flow of their diagram only, not according to an arbitrary key. (See figure 3)

## PHYSICAL REQUIREMENTS

The hardware and software features of the PLATO system fulfilled our three basic requirements for a system to create student-controlled graphics editors.

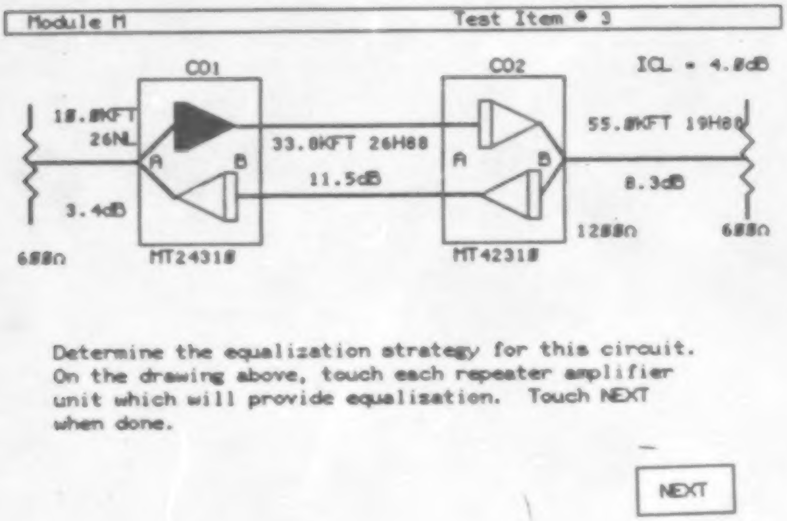
- First, it uses a "bitmapped" display--each dot on the screen can be turned or or off individually without affecting the rest of the screen. Without this it would be impossible to create complex graphic displays and locate them properly.
- Second, a cursor can be provided which the student can move to any location of the display.
- Third, flexible software means of creating and manipulating graphics displays are provided.

## SUMMARY

What we have done is only a start. We have taken only a few of the possible paths. But it can be done, and we would no longer consider developing any computer-based courses without using the graphics capabilities we have developed.

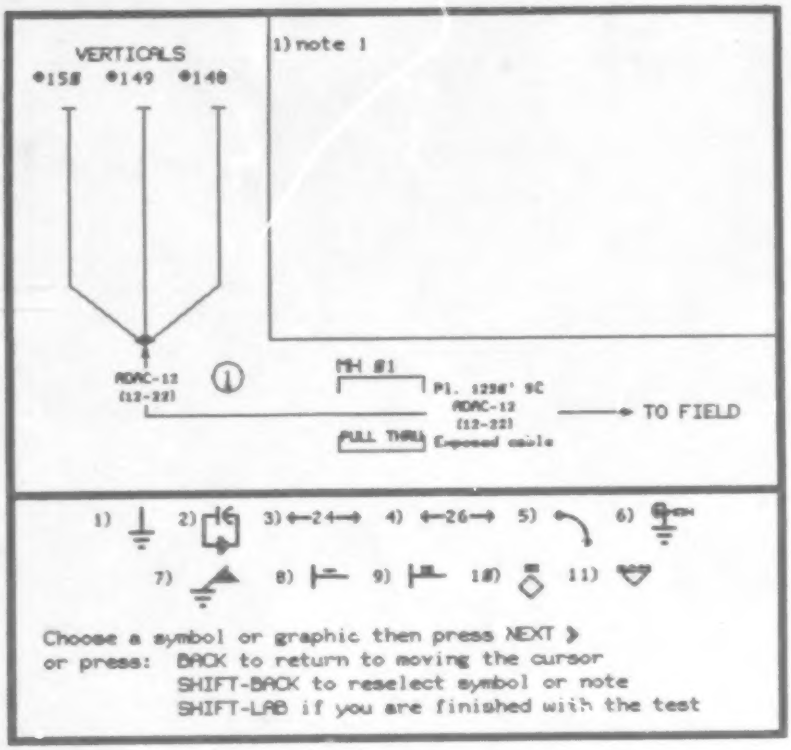
## REFERENCES

Gagné, R.M., The Conditions of Learning, 2nd ed. New York: Holt, Rinehart & Winston (1970).


Module M
Test Item # 3
CO1
CO2
ICL = 4.0dB
10.0KFT
26NL
A
B
33.0KFT 26H88
11.5dB
A
B
55.0KFT 19H88
8.3dB
3.4dB
600Ω
MT24310
MT42310
1200Ω
600Ω
NEXT

Determine the equalization strategy for this circuit. On the drawing above, touch each repeater amplifier unit which will provide equalization. Touch NEXT when done.

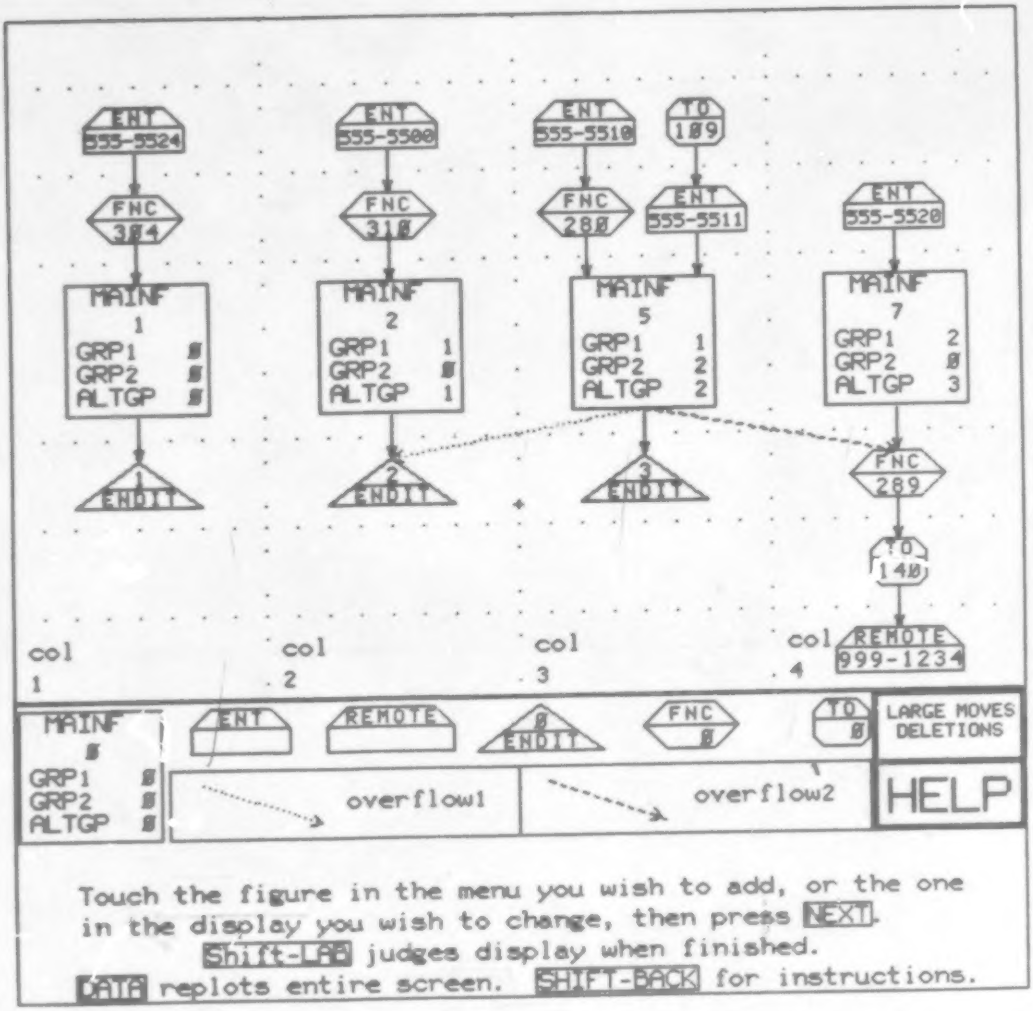
FIGURE 1


VERTICALS
#150 #149 #148
1) note 1
ADAC-12
(12-22)
①
MH #1
P1. 1230' 9C
ADAC-12
(12-22)
TO FIELD
PULL THRU
Exposed cable
1)
2)
3) ←24→
4) ←26→
5)
6)
7)
8)
9)
10)
11)

Choose a symbol or graphic then press NEXT ›
or press: BACK to return to moving the cursor
SHIFT-BACK to reselect symbol or note
SHIFT-LAB if you are finished with the test

FIGURE 2

STUDENT EDITED GRAPHICS


ENT
555-5524
FNC
304
MAINF
1
GRP1 0
GRP2 0
ALTGP 0
1
ENDIT
ENT
555-5500
FNC
310
MAINF
2
GRP1 1
GRP2 0
ALTGP 1
2
ENDIT
ENT
555-5510
TO
109
FNC
280
ENT
555-5511
MAINF
5
GRP1 1
GRP2 2
ALTGP 2
3
ENDIT
ENT
555-5520
MAINF
7
GRP1 2
GRP2 0
ALTGP 3
FNC
289
TO
140
REMOTE
999-1234
col 1
col 2
col 3
col 4
MAINF
0
GRP1 0
GRP2 0
ALTGP 0
ENT
REMOTE
0
ENDIT
FNC
0
TO
0
LARGE MOVES
DELETIONS
overflow1
overflow2
HELP

Touch the figure in the menu you wish to add, or the one in the display you wish to change, then press NEXT.
Shift-LAB judges display when finished.
DATA replots entire screen. SHIFT-BACK for instructions.

FIGURE 3

# AN IMPLEMENTATION STRUCTURE FOR INSTRUCTIONAL COMPUTING

Eileen K. Gress

Fairport Central Schools, New York

## ABSTRACT

A five stage development cycle is presented, based on a theory described by Wood, Thompson and Russell, as a model for implementing an instructional computing program. The model presented includes Readiness, Training, Implementation, and Evaluation. This is juxtaposed with the author's practical experience of implementing a K-12 instructional computing curriculum in Fairport, New York. Events in each stage are related to organizational requirements for resource allocation, including grouping, spatial arrangements, personnel to perform new roles, new materials, and allocation of time and space; new role relationships, strategies and tasks; and staff development. These organizational requirements are differentiated for each of five categories of instructional computing; the computer as an object of study, as a tool for solving problems, as an instructional delivery system, as a management tool, and as a machine to be programmed. Particular attention is paid to requirements for resource allocation and staff development. A model inservice plan is suggested.

## INTRODUCTION

"GIGO" influences the world of educational planning much as it affects the world of computer technology. The saying "Garbage In, Garbage Out" should be heeded by every planner of instructional computing. Without a properly conceived and executed implementation plan, computing programs will smack of the same disorganization that has turned other educational innovations from fads into follies. A five stage development cycle, including Readiness, Planning, Training, Implementation, and Evaluation, can be used to outline an effective implementation plan.

The development cycle is based on a staff development paper compiled by Wood, Thompson and Russell (1981). The cycle defined by the authors comprises Readiness, Planning, Training, Implementation, and Maintenance.

Traditional stage models of development primarily attend to the written product (see for example, Schutz, 1970). The tendency to be narrow minded, however, must be overcome if computer technology is going to have any major effects on schooling. Implementation planners must be mindful of organizational changes other than curriculum. Fullan and Pomfret (1977) suggest four other dimensions that must be attended to: resource allocations, including grouping, spatial arrangements, personnel to perform new roles, new materials, and allocation of time and space: new role relationships, strategies, and tasks; knowledge and understanding that the users have about the innovation's components; and value internalization. I shall address each of these concerns through the use of the development cycle outlined above.

## READINESS STAGE

The concern of the Readiness Stage is with creating an environment that augurs well for successful program implementation. Specifically, two broad goals can be outlined at this stage: awareness of the components of instructional computing, and the development of a positive belief system toward computing, including a desire to learn about the subject, confidence in one's ability to learn to use a computer, and beliefs that the computer is both an effective teaching tool and that knowledge about computing is a worthwhile goal.

Three variables must be considered at this stage: the computing advocate(s), the various audiences of a school district, and the activities used to reach the goals of this stage. The advocates are clearly change agents, and may act as facilitators, trainers, and resident experts. Though the move toward instructional computing may be a grass roots or administrative effort, care must be taken to address each major public of the district.

Faculty must be fully aware of the components of computer education. Each school is considered a separate entity, and a school environment that encourages mutual respect and open communication is an important prerequisite for a successful innovation (Wood, et al., 1981). That is, faculty must be aware of the components of computer education and its effect on their job performance. This enables the innovation to become personally meaningful (Miller & Wolf, 1978, p. 142).

A faculty awareness workshop should have three broad goals: to develop knowledge of what an instructional computing program can consist of, to create awareness of potential changes in the role of the teacher, and to develop a positive attitude toward the subject. A number of taxonomies exist for instructional computing (see, for example, Salisbury, 1973; Dickerson & Pritchard, 1981); however, any model used must correspond closely with possible curriculum and staff development goals. I suggest that the following categories characterize the instructional uses of computers:

. the computer as an object of study
. the computer as a tool for solving problems
. the computer as an instructional delivery system
. the computer as a management tool
. the computer as a machine to be programmed

Staff members must be aware of the differences between these categories. The first, the computer as an object of study, implies learning about computers qua computers. It may include the following topics: history of computers, organization and operation of computers, social implications, careers, applications, and interaction with the computer (Fairport Central School District, 1981).

Using the computer as a tool for problem solving may include employing the computer as a calculator, retrieving information from data bases, and analyzing data using graphing programs. Student requisites for this application of the computer include the ability to define a problem, develop a solution model, and use appropriate software.

Computer-assisted instruction (CAI) is the term most often used when delivery of instruction is considered. A taxonomy of CAI includes drill and practice programs, tutorials, and simulations.

Management of instruction includes diagnosis and testing, student record keeping, and instructional material generation. The computer's capacity for dealing with each student as a unique individual may revolutionize the role of the teacher.

Programming requires not only knowledge of a programming language, but the ability to apply problem solving strategies. The student programmer must be able to define the problem, structure a solution strategy, and evaluate the efficiency and effectiveness of that strategy.

Once basic definitions are presented to staff members, workshops should enable teachers to view software and to use the machine. A good workshop leader fosters the development of the following attitudes:

. use of the computer as an aid to instruction can increase teacher effectiveness
. knowledge about computing is a worthwhile student goal
. personal knowledge about computing and skill at using the computer is desirable

District administrators should also develop an awareness of computing. The support of the building administrator is critical and must be generated at the outset. The principal's contribution lies in his/her providing the project with "legitimacy" by supporting it (Berman and McLaughlin, 1978, p. viii). At this stage, he/she should show support for the potential program by encouraging staff members to participate in awareness workshops, graduate courses, etc. Furthermore, administrators should actively participate in such workshops. Evidence from Rand's study of federal programs supporting educational change (commonly called the Change Agent Study) suggests that interest shown by administrators clues the teachers in to how hard they should work (McLaughlin & Marsh, 1978).

Awareness workshops should also be aimed at school board members, parents, and students. Wood et al. (1981) suggest that support for a new program is obtained through the broad-based involvement of teachers, building and central office administrators, school board members, and parents. The activities and focus of the Readiness Stage are summarized in Figure 1.

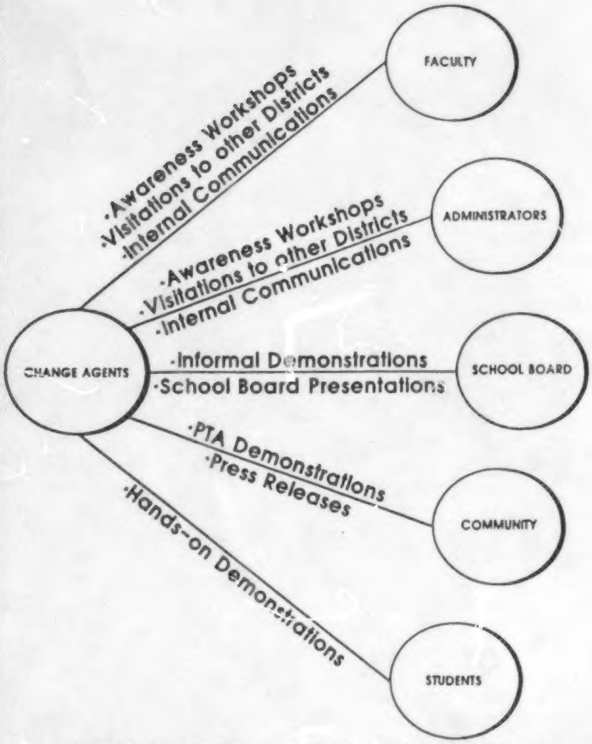


**FIGURE 1. Activities of the Readiness Stage**

Let's take a case in point. The concept of a multi-level computer education program originated in Fairport, New York, as the idea of one individual. As a result of small group faculty discussions and presentations to students, the number of computer education "converts" increased. At that point, a campaign to involve the administration and community ensued. Hands-on presentations were given to the board of education, to the parent-teacher's association, to individual faculty members, to groups of students, to building administrators, and to parents' groups. At no time were these meetings made mandatory. News articles appeared in local papers. The support of the community was thus gained, enabling the district to take the next step in introducing computer education at all levels.

It is at this stage that a framework for long range planning should be developed. The Change Agent Study found that ambitious projects were either met with indifference, resistance, or short life if the planning was either grass roots or district office dominated. Thus, a broad-based committee, including faculty members and administrators, should address the tasks of goal setting, resource allocation, staff development, curriculum development, program management, and evaluation.

A suburban district, Fairport comprises nine schools with a total enrollment of 7500. The Computer Curriculum Committee includes a teacher or department chairperson representative from each building and a district administrator who acts as chairperson. Membership on the committee is voluntary.

The committee meets monthly, and has developed plans for curriculum development, planned and executed faculty orientations for the completed curriculum, written a software evaluation form, and addressed major implementation issues. As an ongoing activity, short and long range plans are developed.

## PLANNING STAGE

The most fundamental outcome of this next stage, Planning, is the development of specific long range program goals. These goals should be broad and ambitious, rather than idiosyncratic, and should appeal to a teacher's sense of professionalism (McLaughlin & Marsh, 1978). They should be constrained by the results of a needs assessment (administrative, teacher and student needs) and by realistic expectations for resource allocation.

Each form of instructional computing should be examined as it relates to district needs, resource allocation and staff development. That the computer should be an object of study for all students has been recommended by various organizations and individuals (see, for example, NCTM, 1980; Gress, 1981).

Introduction of a strand where the computer is studied requires minimal changes in organizational structure. Existing staff members may deliver instruction after appropriate staff development. The strand may be incorporated into a single subject, such as mathematics or science, or may be woven into both the sciences and humanities. Though acquisition of instructional materials is desirable, computer hardware is not necessary. Moreover, neither the role of the teacher nor student grouping patterns need change. To introduce other forms of computing requires greater resource allotment.

The computer becomes a tool for solving problems when it is used to retrieve data from a file, to demonstrate examples of a concept to be developed through inductive reasoning or to process data by generating graphs, lab reports, or business reports. Appropriate applications change by subject and grade level, and so do the costs of implementation. While one computer and minimal staff development might be sufficient for the elementary school, departments such as business and the library might need substantial investments in hardware and training. Depending on the application, activities may be oriented toward individuals or small groups.

The use of computer-assisted instruction may involve major expenditures, changes in teaching strategies, and management considerations. Since a program to utilize computer-assisted instruction (CAI) can range from occasional classroom use of software to the use of individualized instruction for each student in a class, a thorough needs assessment should determine the extent of CAI desirable.

Research on CAI suggests that it is most effective with slow students (DeVault, 1980). The quantity of time at the machine is as critical as the quality. Ten minutes a session for three times weekly is minimal for adequate achievement gains. For such a program, instructional management capabilities are critical. Major organizational changes are made in the role of the teacher, grouping patterns, staffing, and budget allocations.

Instruction in programming requires a large outlay in staff development. Decisions relative to grade levels and number of teachers to be affected can reduce costs; however, they must be anticipated at this stage. Papert (1980) has shown that the use of a suitable language can enable students to program as early as grade four. Furthermore, a low computer to student ratio increases the active, hands-on learning necessary for internalization of new concepts. Papert's experimental work with LOGO utilizes a 1 to 4 ratio (Papert, Watt, et al., 1979).

Efficient program management is crucial to goal achievement. The Fairport Schools consider hardware location and accessibility, security of equipment, supervision and scheduling. Parent volunteers and existing personnel have been trained to duplicate software, supervise students, schedule hardware use, file and distribute software, and catalog instructional materials.

Curriculum development should be an outgrowth of program goals. A variety of curricula for learning about computing have been written. The Fairport Central Schools (1981) use a K-12 curriculum which includes careers, components of a computer, applications, social implications, flowcharting, history, and interaction with a computer. This curriculum is used by all students. In addition, programming strands for gifted and talented students as well as secondary school computer science electives are in place.

Curriculum plans should be tempered by factors other than resource allocation. The level of computer literacy of the staff may make a long range implementation schedule desirable. The impact of new curriculum on traditional instructional areas must also be considered. Bear in mind, however, that what was 'basic' in a low technical era may no longer be basic.

## TRAINING STAGE

With a curriculum defined, staff development objectives can be formulated and resources allocated. Objectives should address knowledge, values, teaching strategies, and new teacher roles.

Though this is a sizable task, curriculum and program management should now be clarified. The next stage, Training, focuses on the staff development necessary for program implementation.

Of the many issues involved in staff development, those that stand out are format, site, instructor, and delivery style. A workshop format delivered over a span of time is most effective (Berman & McLaughlin, 1978). Though the number of contact hours depends on instructional goals, I have found that an introductory workshop of 12-15 hours is effective. Though choice of site is less important than the time element, it can affect teacher attitude and ability to integrate new skills. In general, teachers prefer on-site delivery (Yarger et al., 1978) and are slightly more successful in improving teaching skills (Cruickshank et al., 1979) in such situations. Regarding who shall provide instruction, the use of inside personnel or outside consultants to train key personnel is recommended (Wood et al., 1981). Finally, programs that individualize instruction are more successful than programs that use common activities for all participants (Cruickshank, et al., 1979).

The Fairport Schools use a hands-on group problem solving approach as the basis for its staff development. Two levels of fifteen hour inservice courses, introductory and intermediate, are offered. In addition, short building-based workshops, with topics defined by teachers or administrators, are provided. Peer support, encouraged by small group problem solving, enables participants to assimilate new skills, behaviors, and attitudes. The process of inservice models the changes desired of the participants, making the material immediately applicable (see Figure 2).

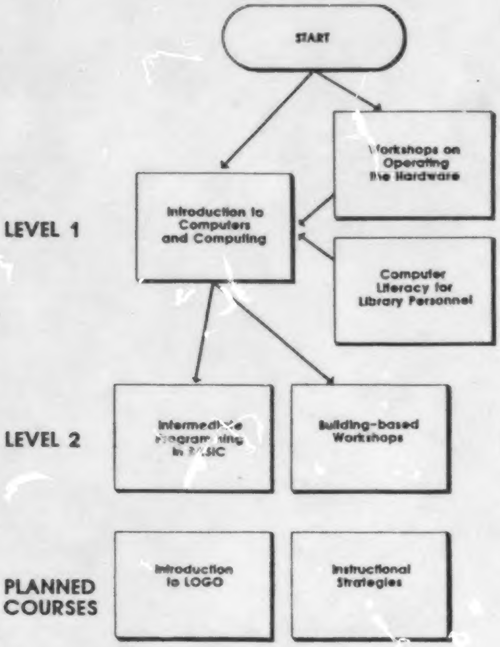

START
Workshops on Operating the Hardware
LEVEL 1
Introduction to Computers and Computing
Computer Literacy for Library Personnel
LEVEL 2
Intermediate Programming in BASIC
Building-based Workshops
PLANNED COURSES
Introduction to LOGO
Instructional Strategies

**FIGURE 2. Instructional Computing Inservice in Fairport, New York**

This training stage ties together much of the planning completed in the earlier stages. Faculty should be able to implement the curriculum and management tasks. Staff development, however, should continue into the fourth stage, when the instructional computing program is implemented.

## IMPLEMENTATION STAGE

The thrust of the Implementation Stage is"...to provide an environment that will support the transition of inservice learnings from the workshop setting into the daily activities of classroom teaching..." (Wood et al, 1981, p. 82). New teaching strategies and roles are best rooted when on-the-job assistance is available. This is particularly appropriate when major efforts in computer assisted instruction or programming instruction require new skills in managing an individualized program.

The Fairport Schools provide such an environment in two ways. An ongoing committee is used as a communicational link among buildings. A representative from each building is available to assist teachers in implementing the curriculum. Second, short workshops are provided by a district mathematics teacher. These workshops are held at the building level and topics are defined based on needs expressed by staff members. The first workshop was used to review hardware operation and discuss application of specific software to classroom use.

## EVALUATION STAGE

With a program in place, we educators often conveniently omit the evaluation stage. Yet it is primarily this last stage which "establishes continuous monitoring to determine whether new behaviors are still being practiced and goals met" (Wood, et al., 1981, p. 86).

Though achievement of goals is a primary research question, it does not become a question until the level of implementation is determined. Investigating whether the program is being implemented with fidelity can turn up flaws in the implementation process. Perhaps teacher attitude is poor and is causing a low degree of implementation. Or lack of on-th-job assistance may prevent teachers from adopting new roles. Formative evaluation should be used at each stage of the process. When the program is in place, staff, students, and parents should be queried to determine the amount of congruence between stated objectives and actual instruction. Open ended interviews can be used to determine the source of any difficulties.

Instruction in computing may have an overall school effect. Measures should be made to determine changes in student or teacher motivation. Perking one's ears up to overhear hall conversations is as valid a measure as recording attendance figures.

I have reviewed five stages as if they were discrete processes. Indeed they are not. Rather, they present a convenient way to describe and plan a process. In reality, the stages overlap and should be revisited. Nevertheless, they aid the educational planner by providing a set of considerations to help guide decision making. Instructional computing has characteristics that are unique to that innovation. We cannot enter that world blindly.

## NOTES

[1] Writers of the Fairport Computer Curriculum were Bill Kling (chairperson), Nancy Cummings, Chris Martin, Mike Motkowski, Linda Pearles, Laurie Rosborough, and Jean Waganka.

[2] The staff development programs have been created by Fairport Personnel: Linda Pearles, Laurie Rosborough, Stella Astheimer, and Eileen Gress.

## REFERENCES

Berman, P., & McLaughlin, M. W. Federal Programs Supporting Educational Change, Volume viii: Implementing and Sustaining Innovations. Santa Monica, California: Rand Corp., 1978. (ERIC Document Reproduction Service No. ED 159 289).

Cruickshank, D. R., Lorish, C., & Thompson, L. What We Think We Know About Inservice Education. Journal of Teacher Education, 1979, 30, 51-54.

DeVault, M. V. Computers. In E. Fennema (Ed.), Mathematics Education Research: Implications For The 80's. Reston, Virginia: Association for Supervision and Curriculum Development, 1981.

Fairport Central School District. K-12 Computer Education Curriculum, Fairport, New York: Fairport Central Schools, 1981.

Fullan, M., & Pomfret, A. Research on Curriculum and Instruction Implementation. Review of Educational Research, 1977, 47(2), 335-397.

Gress, E. K. The Future of Computer Education: Invincible Innovation or Transitory Transformation? The Computing Teacher, 1981, 9, 39-44.

McLaughlin, M. W., & Marsh, D. D. Staff Development and School Change. Teachers College Record, 1978, 80, 69-94.

Miller, L., & Wolf, T. E. Staff Development for School Change: Theory and Practice. Teachers College Record, 1978, 80, 59-68.

National Council of Teachers of Mathematics. An Agenda For Action. Reston, Virginia: NCTM, 1980.

Papert, S. Mindstroms. New York: Basic Books, Inc., 1980.

Papert, S., Watt, D. et al. Final Report of the Brookline LOGO Project. AI Memo No. 545, LOGO Memo No. 53. Cambridge: MIT AI Lab, 1979.

Schutz, R. E. The Nature of Educational Development. Journal of Research and Development in Education, 1970, 3, 52.

Yarger, S. J., Howey, K. R., & Joyce, B. Inservice Teacher Education. Palo Alto, California: Booksend Laboratory, 1978.

Wood, F. H., Thompson, S. R., & Russell, F. Sr. Designing Effective Staff Development Programs. In B. Dillon-Peterson (Ed.), Staff Development/Organization Development. Alexandria, Virginia: Association for Supervision and Curriculum Development, 1981.

# MENO-II: An AI-CAI Programming Tutor

Elliot Soloway *
Eric Rubin **
Beverly Woolf **
Jeffrey Bonar **

*Department of Computer Science
Yale University
New Haven, Connecticut 06520

**Department of Comp. & Info. Science
University of Massachusetts
Amherst, Massachusetts 01003

## Abstract

MENO-II is a computer-based tutor for novices learning to program in Pascal. It can analyze simple looping programs and identify 18 types of nonsyntactic bugs. A bug is tied to a set of possible misconceptions which the student might have. Currently, MENO-II simply outputs the list of bugs it has found, along with the associated misconceptions.

MENO-II uses techniques from Artificial Intelligence in order to analyze a program into its deep structure, functional components, and to represent explicitly programming knowledge. MENO-II has built into it a catalogue of common bugs, which has been constructed from empirical research we have done with novice programmers. We suggest that limited AI-CAI systems, of which MENO-II is an example, can be the next generation of CAI systems --- in the near future.

This work was supported by the Army Research Institute for the Behavioral and Social Sciences, under ARI Grant No. MDA903-80-C-0508. Any opinions, findings, conclusions or recommendations expressed in this report are those of the authors, and do not necessarily reflect the views of the U.S. Government.

## 1. Introduction

We all know -- and grumble -- about the error messages that one gets from a computer system. That problem is magnified for the novice programmer who is learning a first programming language. With the number of individuals who are learning programming rapidly growing, there is a clear need for intelligent tutoring systems which can assist the novice at a most critical time: when he/she is alone, one-on-one, with the beast.

## 2. Using Artificial Intelligence Techniques

There are two major differences between MENO-II and typical frame-based CAI systems:

1. In contrast to more classical CAI systems which build the subject matter into frames which are tied together with a branching strategy, the knowledge about programming which MENO-II has is represented explicitly, in the form of a network, and the instructional strategy is independent of that knowledge and also encoded explicitly. Thus, MENO-II generates what to say based on the particular situation.

2. In contrast to more classical CAI systems which respond to only a small set of student errors, MENO-II can cope with 18 different types of program bugs. These bugs are tied explicitly to a knowledge base of potential misconceptions, which is accessed when interacting with the student.

In order to build such a system, we have used many techniques which have been developed in Artificial Intelligence. For example, the knowledge representation we use is called KL-ONE, and was developed as a general knowledge representation language (Brachman, 1978). See Clancey (Clancey, 1979) and Goldstein (Goldstein, 1979) for a more extensive discussion of the role which AI can play in CAI.

## 3. Objectives and Status of System

MENO-II is a tutoring system designed to help novice programmers learn Pascal. The goals of MENO-II are:

- catch run-time (semantic and pragmatic) bugs in the student programs[1]
    * focus on introductory programs: straight line, branching, simple looping.
    * use MENO-II in conjunction with an existing lecture course on Pascal.
- suggest misconceptions in the students' heads which underlie the bugs
    * "talk" in language which is close to programming so student will understand
- instruct/tutor the student with respect to the misconceptions.

The current status of the system corresponding to these objectives is:

- MENO-II can find 18 bugs with respect to repetition and assignment
- MENO-II can suggest underlying misconceptions for those bugs
- only a rudimentary form of tutoring has been implemented in MENO-II.

MENO-II is divided into two major components: The Bug Finder, and The Tutor; each will be discussed in turn. We close with a discussion of plans for MENO-II's expansion.

## 4. Leverage on Program Understanding: The Bug Finder

The Bug Finder must be able to recognize two types of bugs: problem independent ones (semantic bugs) and problem dependent ones (pragmatic bugs). An example of the former type is the explicit inclusion of an increment to the index variable in a for loop. Bugs of this sort reflect confusion about the semantics of the various programming language constructs.

In order to recognize problem dependent bugs, however, the Bug Finder must be told what the program is supposed to do; oftentimes a student's program will run, but it will solve the "wrong" problem. For example, in Figure 1a, we depict a problem for which the program in Figure 1b is the correct solution; the program in Figure 1c does not solve the problem, but does in fact execute. In other programming tutors, the student provided a specification of the goals of the program (e.g., (Goldstein, 1974, Miller, 1978)). In contrast, our approach is to have the teacher provide the specification, since everyone in the course will be working on the same problem.

Currently, the specification of the intended program is in terms of bugs typically found in attempts at the correct program. That is, we have built up a catalogue of common bugs in simple looping programs based on a number of

---

[1] While syntax errors are clearly troublesome for novices, we believe that systems like structure editors (e.g. the Cornell Program Synthesizer (Teitelbaum and Reps, 1980)) will soon be available, and will help facilitate the creation of syntactically correct programs.

Problem: Read in numbers, taking their sum, until the number 99999 is seen. Report the sun. Do not include the final 99999 in the sum.

(a)

```
PROGRAM CORRECT-EXAMPLE;
   VAR  SUM, NEW : REAL;
   BEGIN
     SUM:=0;
     READ(NEW);
     WHILE NEW <> 99999 DO
        BEGIN
        SUM:=SUM + NEW;
        READ(NEW)
        END;
     WRITELN(SUM)
   END;
```

A correct version of a program which solves the above problem.

(b)

```
PROGRAM BUGGY-EXAMPLE
   VAR SUM, NEW : REAL;
   BEGIN
      WHILE NEW <> 99999 DO
         BEGIN
         READ(NEW);
         SUM := SUM + NEW
         END
      WRITELN(SUM);
   END;
```

A buggy program, which is an attempt at solving the above problem. The variable SUM is not initialized to 0. Similarly, the variable NEW has no value the first time it is tested at the top of the WHILE loop. Finally, notice that this program will add in the final 99999 into SUM.

(c)

Figure 1: Sample Programs

empirical studies (e.g., (Soloway et al., 1981a, Soloway et al., 1982a, Soloway et al., 1981b)). This catalogue is built into the Bug Finder. Preliminary data analysis suggests that the class of 18 bugs for problems like those in Figure 1c and Figure 2 covers 70% of all (non-syntactic) student errors.

The Bug Finder, which is written in Pascal works by first analyzing a student's program into a "deep structure" representation (Brown, Collins, and Harris, 1978), and then matching that analysis against the bugs in the Bug Catalogue. The deep structure representation of the program specifies the functional characteristics of the program. The set of primitives of this deep structure representation is based on what an expert programmer might know about problems of this type. In particular, this knowledge is focused on types of <u>looping plans</u> and the various <u>roles</u> which variables play in programs. For example, the program in Figure 1b illustrates what we call the New_Value Controlled Running_Total Loop Plan; it is just a special case of the Running_Total Loop Plan, in which the loop, while accumulating a total, is controlled by the value of the <u>read</u> variable (i.e., the New_Value Variable). This knowledge is described at greater length in (Soloway et al., 1982a), and draws on the work of (Miller, 1978, Rich, 1980, Rich and Shrobe, 1978, Waters, 1979, Soloway and Woolf, 1979, Barstow, 1979).

There are a number of stages of analysis that the Bug Finder performs on the student's program in order to derive its deep structure representation. The objective is to move from a standard parse tree to a parse tree which is decorated with comments. Such annotations reflect the plans and bugs which were identified. For example, one class of bugs involves variable plans which could not be fully matched, e.g., the program in Figure 1b fails to the Running-Total Variable, SUM. Other bugs are caused by incorrect ordering of the operations in the loop. For example, the program in Figure 1c will incorrectly add the sentinel value 99999 into the running total; this is a classic example of the "off by one" bug. Moreover, upon entry into the loop, NEW is undefined. A correct version of this program, as also shown in Figure 1c, would READ a value into NEW before the loop, and would perform the READ on NEW in the loop after the running total update.

The process of matching plans to programs, and of identifying bugs in the plans is clearly a heuristic one, and is not guaranteed to be 100% successful. While there are techniques such as data and control flow analysis which are exhaustive, they are quite low level in comparison to the plan-based analysis. Thus, it is not clear that they would even be of much use in programs of the complexity used in introductory programming courses. While we are not adverse to adding such analysis techniques, we believe that the real power of the Bug Finder will come from building up more extensive and and richer libraries of plans (Rich, 1980) and bugs.

As we said, the Bug Finder is currently specifically programmed to identify a particular set of bugs. In order to make the Bug Finder more flexible, we are looking into making it more table driven. That is, a better procedure would be to describe the bugs which should be searched for in a high level language and have the computer automatically compile these descriptions into bug finding procedures. Based on our empirical studies of novice programmers, and on our plan language, we are designing just such a Bug Description Language.

## 5. The Tutor: Inferring Misconceptions from Bugs

Given that the Bug Finder has found a bug (or bugs) in a student's program, the next step is to hypothesize the set of potential misconceptions which were in the head of the student which might have been responsible for the program bug. In particular, the Bug Finder passes the number of the bug to the Tutor, and it is the job of the Tutor to perform this misconception analysis. Currently, the Tutor hypothesizes the misconceptions and simply reports them, plus the correct concepts, to the student; the Tutor does not attempt to engage in a dialogue with the student.

There are four knowledge bases currently in the Tutor.

The Competence Model -- the correct knowledge about programming is contained in this component.

The Bug Network -- the common bugs we have identified in our empirical work are catalogued in this component.

The Misconception Network -- associated with a bug(s) are misconceptions which could give rise to the bug; this network explicitly stores the misconception(s) and a tutorial associated with each bug.

The Performance Model -- this is the system's hypothesis as the what the student does and does not know, the student's history of interaction, etc.

In what follows we will illustrate how the above knowledge bases are used by the Tutor to produce its misconception analysis and tutorial output.

## 6. Examples of the Tutor's Analyses and Interactions

The Tutor, which is written in LISP, currently has two modes of operation: Full Comment or Partial Comment. In the latter mode, only errors which the Tutor deems as "serious" will cause the Tutor to say something to the student. The objective here was to be less intrusive. In the former mode, any semantic or pragmatic bug will elicit a response from the Tutor.

Consider first the tutoring session depicted in Figure 3. In particular, this program was an attempt to solve the problem in Figure 2; a correct program to this problem is also given. While the correct program reads a value into the New_Value Variable (NEW in Figure 2), the program in Figure 3 instead increments the New_Value Variable (POSEIDON) by 1. Our empirical studies have shown this to be a typical bug.

Running in Full Comment mode, the Tutor first prints out its analysis of the roles which the variables play in the program, e.g., POSIDEN is the New_Value Variable, COUNT is the Counter Variable, etc. Next, both the bug and the correct action are described to the student. Bugs, indexed by their numbers, are stored in the Bug Knowledge base, and are tied to the Competence Model by Buggy_Version links. For example, in

Problem: Read in a set of integers and print out their average. Stop reading numbers when the number 99999 is seen. Do NOT include the 99999 in the average.

```
PROGRAM CORRECT-EXAMPLE;
  VAR
    TOTAL,NEW,COUNT:INTEGER;
    AVE:REAL;
  BEGIN
    TOTAL:=0;
    COUNT:=0;
    READ(NEW);
    WHILE NEW <> 9999 DO
      BEGIN
        SUM := SUM + NEW;
        COUNT:=COUNT+1;
        READ(NEW)
      END;
    AVE:=TOTAL/COUNT;
    WRITELN('THE AVERAGE IS ',AVE)
  END.
```

Figure 2: Problem and Program Solution

Figure 4 we see that Bug 205 is associated with the New_Value Variable. The notation indicates that the correct way to initialize and update a New_Value Variable is via a read, and that Bug 205 indicates that the update was actually accomplished via an assignment statement. The Tutor generates the English description of the bug from the Bug Network representation, and generates the English description for the correct modification technique from the Competence Model.

Associated with a bug is a set of possible misconceptions which the student might have and which could cause the observed bug. This information is stored in the Misconception Network. In Figure 4 we see that there are two possible misconceptions associated with Bug 205 (the bug in the program in Figure 3): the Read Declaration Misconception and the Over-generalize Counter Misconception. The pre-specified text associated with each one of these misconceptions is then displayed to the student. In the next version of the Tutor, a series of questions will be generated and asked of the the student,

```
PROGRAM AVERAGE1(INPUT,OUTPUT);
  VAR
    SUM,POSIDEN,COUNT:INTEGER;
    AVE:REAL;
  BEGIN
    SUM:=0;
    COUNT:=0;
    READ(POSIDEN);
    WHILE POSIDEN<>9999 DO
      BEGIN
        SUM := SUM + POSIDEN;
        COUNT:=COUNT+1;
        POSIDEN := POSIDEN + 1;
      END;
    AVE:=SUM/COUNT;
    WRITELN('THE AVERAGE IS ',AVE)
  END.
```

POSIDEN is the New Value Variable
COUNT is the Counter Variable
SUM is the Running Total Variable

You modified POSIDEN
by adding POSIDEN to 1
where as...
you should modify the
New Value Variable
by calling the READ
procedure: READ(POSIDEN).

Two misconceptions can be associated with this bug:
1. You might be thinking that the single call to the READ procedure [READ (POSIDEN)] at the top of your program is enough to define a variable which will always be read in from the terminal. In fact you need to call the read function a second time in your program to read in additional values. 2. You might be thinking that POSIDEN is like COUNT, in that adding 1 to a variable will retrieve its next value. The computer does not know to reinterpret + 1 in the former case to be a READ.

Figure 3: Example of MENO-II's Analysis

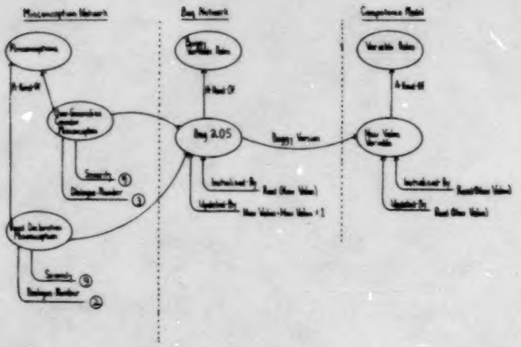


Figure 4:

A Slice Through Three Knowledge Bases

The nodes and arcs in this figure represent a small fragment of the actual knowledge bases. Also, the notation has been simplified to facilitate readability.

in order to help differentiate between the competing hypotheses. Currently the Performance Model is simply a record of the bugs and misconceptions constructed by MENO-II. Eventually, more history will need to be kept and used in the diagnosis of a student's misconceptions.

## 7. Concluding Remarks

Clearly, the MENO-II has quite limited functionality. We plan to significantly extend its capabilities along several dimensions.

1. The range of bugs and misconceptions which MENO-II is capable of coping with needs to be enlarged. The keys to this task are: (1) the continued development of a theory of programming knowledge (Soloway et al., 1982a, Soloway et al., 1982b) and (2) empirical studies which seek to evaluate that theory and to identify additional bugs and misconceptions. In fact, we have used the Bug Finder as a tool in our empirical work. We have collected "on-line protocols" which are copies of all syntactically correct programs generated by students at the computer terminal. The only way to analyze this volume of data is to use the computer; we are running the Bug Finder over this data, and counting the types and frequencies of the various programming bugs (Bonar, Ehrlich, and Soloway, 1981).

2. The ability to more accurately diagnose misconceptions is very important. Currently, only evidence from the program being analyzed is used in this process. We plan to have the Tutor engage in a limited question-answer dialogue with the student in order to gather more information upon which to base its diagnosis. The student will not answer in natural language, but rather, will chose among a set of alternatives. Also, the history of the students' interactions will be kept and used in the analysis. By watching the pattern of bugs and patches which the student makes to the program, the system will be able to build up a better model of the student's understanding.

3. The question-answering interaction serves another purpose: it combines features of a Coaching strategy (Goldstein, 1976) with features of a Socratic Tutoring strategy (Stevens and Collins, 1977). That is, if one observes how a Program Consultant works with novice programmers who are having difficulties with their programs, one sees that oftentimes the students come to understand their misconceptions themselves; the Consultant's role was to ask carefully crafted questions which forced the students to confront their understanding of what they wanted to do with what they did in fact do. We feel

that this paradigm of Consulting is one that is within the state-of-the-art reach of intelligent tutoring systems.

4. We plan to *evaluate* MENO-II in a classroom setting. We have already had a limited test of the Bug Finder in a classroom; it was "hardwired" to handle only 18 bugs and associated misconceptions. Preliminary analysis of a subjective evaluation we carried out after students used the Bug Finder suggests that students found the Bug Finder quite helpful; typically it saved them time, since feedback was immediate and thus they did not have to go and wait in line for a programming consultant, and it gave a detailed analysis of the program's bugs, whereas often the human consultants gave more general advice.

Our intention has been to build a CAI system with enhanced ability to understand and respond to students' buggy programs. In order to realize this objective, we have had to:

1. map out the deep structure knowledge used in programming
2. build a catalogue of common bugs
3. build a program analysis system which can find bugs in programs
4. associate misconceptions with bugs.

We have borrowed quite liberally from Artificial Intelligence and Cognitive Science in order to achieve these goals. MENO-II, however, is restricted: it is not intended to be a self-contained course in programming, capable of generating programs or analyzing all buggy programs, or using a variety of teaching strategies tailored to each student's needs. Nonetheless, we feel that the time is ripe for just such limited AI-CAI systems --- systems with enhanced ability to understand and respond to the answers input by students. Personal computers with sufficient computing resources are becoming available, and the techniques for building AI-based systems are becoming less an art and more an engineering endeavor. Thus, we feel confident that the next generation of CAI systems will incorporate some AI techniques --- and will start appearing soon.

## REFERENCES

Barstow, David. *Knowledge-Based Program Construction*. : Elsevier North Holland Inc. 1979.

Bonar, J., Ehrlich, K., Soloway, E. Collecting and Analyzing On-Line Protocols from Novice Programmers . Behavioral Research Methods and Instrumentation, to appear.

Brachman, R., Ciccarelli, E., Greenfield, N., Yonke, M. *KLONE Reference Manual*. Technical Report 3848, Bolt, Beranek and Newman, Inc, Cambridge, Mass., 1978.

Brown, J.S., Collins, A., and Harris, G. Artificial Intelligence and Learning Strategies. In H.O'Neil (Ed.), *Learning Strategies*, : Academic Press, New York, 1978.

Clancey, W.J., Bennett, J.S., and Cohen, P. *Applications-oriented AI Research*: *Education*. Technical Report STAN-CS-79-749, Computer Science Department, Stanford University, 1979.

Goldstein, I.P. *Understanding Simple Picture Programs*. Technical Report 294, MIT AI Lab, 1974.

Goldstein, I. *The Computer as Coach*: *An Athletic Paradigm for Intellectual Education*. Technical Report 389, MIT AI Lab, Cambridge Ma., 1976.

Goldstein, I.P. The Genetic Graph: A Representation for the Evolution of Procedural Knowledge *International Journal of Man-Machine Studies*, 1979, 1(11), 51-78.

Miller, Mark L. A Structured Planning and Debugging Environment for Elementary Programming. *Int. J. Man-Machine Studies*, 1978, 11, 79-95.

Rich, C. *A Library of Plans with Applications to Automated Analysis*. Technical Report 294, MIT AI Lab, 1980.

Rich, C. and Shrobe, H. Initial report on a LISP Programmer's Apprentice. *IEEE Transactions on Software Engineering*, November 1978, 4(6), 342-376.

Soloway, E. and Woolf, B. *Problems, Plans, and Programs*. Proceedings of Eleventh ACM Technical Symposium on Computer Science Education, 1979.

Soloway, E., Bonar, J. and Ehrlich, K. *Cognitive Factors in Programming: An Empirical Study of Looping Constructs*. Technical Report 81-10, Department of Computer Science, University of Massachusetts , 1981.

Soloway, E., Bonar, J., Woolf, B., Barth, P., Rubin, E., and Ehrlich, K. *Cognition and programming: Why Your Students Write Those Crazy Programs*. NECC, No. Denton, Tx., 1981.

Soloway, E., Ehrlich, K., Bonar, J., Greenspan, J. What Do Novices Know About Pogramming? In Directions in Human-Computer Interactions, B. Shneiderman and A. Badre, Eds., Ablex, Inc. in press.

Soloway, E., Ehrlich, K., Bonar, J. Tapping Into Tacit Programming Knowledge. Proceedings of the Conference on Human Factors in Computing Systems, 1982, ACM, Gaithersburg, Md., in press.

Stevens, A. and Collins, A. *The Goal Structure of a Socratic Tutor*. Technical Report, Bolt Beranek and Newman, Cambridge, Mass., 1977.

Teitelbaum, T. and Reps, T. *The Cornell Program Synthesizer: A Syntax-Directed Programming Environment*. Technical Report 80-421, Cornell University, Dept. of Computer Science, 1980.

Waters, R.C. A Method for Analyzing Loop Programs. *IEEE Trans. on Software Engineering*, May 1979, *SE-5*, 237-247.

# COURSEWARE TRANSPORTABILITY ISSUES

Richard C. Brandt
Video Computer Learning Project
Physics Department
University of Utah
Salt Lake City, Utah 84112

The costs of developing computer-based courseware for education and training are substantial because of the effort involved in both creating and validating the courseware. These costs cannot be amortized unless the courseware is widely used. This will occur only if the courseware can be used on a number of different hardware systems; specifically, if the courseware can be moved, that is transported, to a different system. Unfortunately, this issue has either been ignored or been inadequately addressed for the last ten years. Consider these three major projects: PLATO, TICCIT, and Professor Bork's at U.C. Irvine. Each project used its own language and incompatible display units with the result that little courseware moved from one project to another and the courseware developed at great expense has not been usable on today's commonly used microcomputer systems. With reduced federal support, this luxury can no longer be afforded.

Up to now only CONDUIT has addressed the transportability issue. CONDUIT used the solution that either materials are written in FORTRAN or BASIC for line-oriented terminals or they are written specifically for one of the four common microcomputers: the Apple, the Pet, the TRS-80, or the Atari. This paper suggests an alternative approach to transportability based on two observations: courseware will be moved only if it is easily moved and if it can easily be modified to meet the different needs of new sites.

This paper recommends the establishment of guidelines for hardware, courseware datafile structures, and software to assure the transportability of the next generation of courseware. These guidelines must be designed to encourage rather than discourage the develoment of good courseware. For the purposes of this paper good courseware: (1) is effective and validated; (2) enables teachers to effectively instruct as large a number of students as previously or a larger number; (3) is easily modified by teachers rather than programmers to meet differing needs (either in entry level or values); and (4) is developed by teachers as opposed to programmers. Ideally, programmers will be required only in special cases. As the next generation of microcomputers (based on the new microprocessors: 8086, 68000, etc.) become available it is critical that guidelines are followed to ensure the transportability of the courseware developed.

## HARDWARE

Presently, the biggest obstacle to courseware transportability is associated with a simple hardware issue--the number of lines of text on the screen and the number of characters permitted in each line. PLATO lessons use 32 lines of 64 characters; TICCIT lessons use 16 lines of 43 characters; Bork's Sigma 7 lessons used 35 lines of 74 characters. More recent work (VCIS and Bork's) uses displays that support either 24 lines of 80 characters or 24 lines of 40 characters. Changing between any of these, requires reformatting of the text. Although some of the text can be reformatted by machine translation, it usually requires final hand adjusting, particularly when graphics are interspersed with the text. Even more time consuming is the case where part of the lesson must be rewritten because all the original text cannot be displayed on the screen. Within the computer industry 24 lines by 80 characters is a defacto standard. It is logical to adopt this standard for courseware with two provisions: the top and bottom line on the screen be reserved for special prompts and student information, and direct cursor addressing be provided.

The intermixing of graphics display with text is an essential part of almost any computer assisted instruction lesson or training material. Graphics will be transportable only if (1) a minimum resolution is specified and (2) the shape of the screen is specified. Only a minimum resolution need be specified for graphics since the graphic drivers can easily scale the data as long as the screen has the industry standard shape, that is, a height that is 3/4 the width. If a higher resolution is used is should provide a square aspect ratio (visually a square aspect ratio is more pleasing and the graphic display algorithms are simpler). The suggested guideline is that the display screen take the industry standard shape and that the minimum resolution be 320 × 240.

If color monitors are used, they MUST be high resolution RGB monitors so that text is legible. Ordinary television screens do NOT provide legible text due to the coarse shadow mask.

The ability to intersperse audio-visual material with traditional computer-generated graphics and text is a desirable feature. This is presently available with laser videodiscs. Currently, this material is displayed on a different monitor, but is should be displayed on the terminal screen.

## COURSEWARE DATAFILES

There are several approaches to courseware transportability. One approach, used by CONDUIT, is to use a standard BASIC or FORTRAN and write the material so that it will work on a line-oriented terminal such a Teletype. The disadvantage with this approach is that it cannot support good graphics. Another approach, used recently by CONDUIT and Bork, is to create the lesson for a specific hardward system, and then modify the lesson program so that it runs on another computer. The disadvantage with this approach is that each lesson program has to be modified; this can be very time consuming and expensive. The approach advocated in this paper is that the lesson material be in datafiles rather than in programs. It is very easy to move a datafile from one computer to another. If datafiles are used, only the programs used to edit these datafiles and run the lesson need be modified to run on different computer systems. This is much less costly since only ten to twenty programs need to be modified versus having to modify every lesson.

If the courseware datafile concept is adopted, datafiles will, in the long run, be the most important issue because of the the number of people involved in creating and maintaining courseware. In designing the datafiles, three issues need to be considered: the single versus multiple file system, the editable versus non-editable file question, and file development.

A large single editable file associated with each lesson is undesirable for several reasons. First, software required to edit such a file will probably require a large amount of memory to operate, which may not be present on all systems. Consequently, a multiple file system with separate files for text, graphics, animations, lesson structure, etc. is advantageous because the software required to edit the individual files will require a smaller amount of memory. From the point of view of transportability, this is desirable since it will permit editing on more systems. Second, the multiple file system has the advantage that the updating process is much faster, and requires less disk space if only one file is being changed and backup files are kept on the disk. Third, as better data representations are developed, it is simpler to change only the files that will use the new data representation than to change a portion of one large file. Fourth, with a multiple file system, the individual files will probably be simpler and consequently more robust.

At first glance, it would seem that all files associated with a lesson should be editable. Certainly all display files (graphics, text, and animation files) should be editable. The lesson structure file used by the lesson interpreter program, the program used to present the lesson, should not be editable for two reasons. First, it will necessarily contain a large number of labels and branching information that would be confusing to a non-programmer. Second, permitting editability at this level is likely to produce lesson material that cannot be easily modified. It is important, however, that the lesson structure be editable. This can be accomplished by having a separate editable file called a command file. The command file is converted by a program, in essence a compiler to the non-editable form required by the lesson interpreter program.

Finally the datafile structures are likely to evolve with the development of better algorithms and better representations of data. If the datafiles are carefully designed, the conversions will be easy. Most of the changes can be expected in lesson structure files as new ideas are developed in computer science, particularly in the area of student answer analysis. Similarly, as the necessary concepts are developed in computer science, new datafiles will be developed. Simulations, interactive graphics, and pool type questions are areas where new developments are likely to occur.

## SOFTWARE

Good courseware will not be designed unless educators create and fully control the development of materials. Educators should concentrate on the design of good courseware rather than learning a programming language. There is a very small overlap between excellent teachers and good programmers outside of computer science departments. Consequently, it is essential that software, in this case the interface between the user and the datafiles, be developed that permits educators without programming to create lessons and to modify lessons to meet local needs. Ideally, the software should make the implementation of courseware as easy to use as a good word processor. The three characteristics of good word processors most relevant to courseware implementation are (1) the individuals using the system "get what they see" (2) a hierarchical command structure is used so only a small and largely self-explanatory set of appropriate commands are available at any one time and (3) the software recovers gracefully from user mistakes. Witn software of this type, the educator should be able to easily create on the display screen the text, graphics, and animations that the student will see.

For the majority of users a hierarchical command structure is easier to use. However, very experienced users sometimes find they can create courseware more efficiently with a very broad and powerful set of commands, for example, the TUTOR language used in the PLATO system. Since the lesson is embodied in the datafiles, it makes no difference if the experienced users work with very different software, as long as that software affects the datafiles in the same way.

Sometimes it will be necessary for programmers to add special features to the lesson interpreter program. These special features can be moved efficiently to other machines only if there is a common high level language or appropriate translator programs. Consequently, for transportability, it is essential that the lesson interpreter program is written in a common structured high level language; at present Pascal is the only possible candidate. In addition, the lesson interpreter program must be designed so that is is relatively easy to link specially written features to it.

For machine independence and maintenance, it is desirable but not essential that the software used by educators be also written in a structured high level language; again the only possible candidate at present is Pascal. When Ada becomes sufficiently widespread and reliable, it is likely there will be automatic translators which convert Pascal to Ada (such an effort has already been started at U.C. Berkeley), thus bringing the software into compliance with Department of Defense requirements. However, preliminary data indicates that Ada will require microcomputers having a relatively large address of space, at least 256 kb, which may exclude some systems. It is clear that a small number of machine dependent primitives (file support, rapid graphics, and text display) will, in general be necessary, but the interface to the high level language should be machine independent.

Although a facility for symbolic and algebraic manipulation is highly desirable, it has been rarely used in computer assisted learning. The exception is Suppes' work at Stanford. Such facilities are generally based on the LISP family of languages. A standard portable LISP has been defined by Griss et al. Since LISP requires a very small number of machine primitives, it is relatively easy to implement. However, its use with Pascal is dependent upon the ability of the Pascal to invoke external procedures. In the case where a P-system is being used, it is desirable that LISP be translated into a P-code to enhance the speed of the algebraic and symbolic manipulation. The UCSD P-code is currently inadequate for this purpose. Since LISP systems frequently require large amounts of main memory, support of symbolic and algebraic manipulation will require that the next generation of microcomputers (IBM, WICAT, APOLLO) be used. These microcomputers support an address space considerably larger than that traditionally available on 16-bit microprocessors.

## SUMMARY

With good software, educators will have the opportunity to directly create the courseware without having to contend with programmers as intermediaries. It is critical that the courseware developed in this period be compatible with the rapidly developing microcomputers and computer languages. The suggested guidelines provide a means to achieve this compatibility in a cost-effective manner.

# THE "BUGGY METAPHOR": THEMES IN ALGEBRA AND PRECALCULUS*

Dr. Ronald H. Wenger, Director

Mathematical Sciences Teaching and Learning Center
014 Memorial Hall, University of Delaware,
Newark, DE 19711

## ABSTRACT

Buggy and DEBUGGY are programs developed by J. S. Brown and R. R. Burton which diagnose procedural bugs in basic mathematical skills--integer addition and subtraction. The systematic and comprehensive manner in which this was done for these algorithms and its implications for mathematics learning, instruction, and teacher training provide a metaphor for research and instructional development in other areas of mathematics as well. These themes and others from cognitive psychology and artificial intelligence are having an important influence upon the analysis of mathematics learning and are also beginning to have impact upon the development of instructional materials--both on and off computers. The complexity of the concepts and alogrithms encountered in algebra and precalculus is such that one cannot hope to be comprehensive in the sense that DEBUGGY is. The strategies used there, and also in "intelligent" LISP-based tutors, are simply not efficient or adaptable enough for our purposes (in the short term)--but they serve as very useful paradigms or standards. For those impatient to use the technology to enhance mathematics learning throughout a full course, a somewhat different strategy is needed.

Success in using the technology to improve learning is seriously limited by the lack of suitable research and diagnostic tools to describe students' understanding of algebra and elementary functions. Fortunately, significantly more attention is being given to such topics. The work of Greeno, Anderson, Davis, Resnick, Lewis, Carry, Bernard, Schoenfeld, Matz, VanLehn, Sleeman, and others is providing both results and methods of organizing observations which are very useful.

The projects in the Mathematical Sciences Teaching and Learning Center are heavily influenced by this work. The Center is:

1. Developing means of categorizing conceptual and algorithmic difficulties of the approximately 1400 students who take algebra or precalculus at the University each year. The current database consists of students' performance on approximately 1100 examination problems.

2. Using the database to: identify and classify the most prevalent and important difficulties of students in algebra and elementary functions; and to create sets of problems to diagnose the most prevalent and important classes.

3. Developing methodologies to help determine which conceptual and algorithmic difficulties are "linked" and hypotheses to explain these linkages.

4. Developing a flexible computer program on the University's PLATO system, the Mathematics Interactive Problem Package (MIPP), to support the diagnostic process; several forms of instruction; and research. The MIPP supports, for example, seven modes of questioning; several (contingency) levels of inquiry of students; and branching to appropriate instructional resources.

---

* This work is supported, in part, by National Science Foundation CAUSE Grant Number SER 8103901.

Contemporary Themes in Mathematics Learning Research:

There seems to be a renaissance underway in research on how individuals learn mathematics. This is due, in part, to increased interest in cognitive science, a blend of cognitive psychology and artificial intelligence perspectives and methodologies. Efforts to use the computer and other information technologies in more substantive and powerful ways to enhance learning have certainly been a catalyst for a good deal of this work. The work of Newell and Simon (1972) on human problem solving has, of course, been an important influence.

The entries in the brief bibliography (see that in Rachlin, 1981 also) illustrate the wide variety of investigations into mathematics learning recently. Virtually all of these have been published since 1970. A reference to a useful and readable summary of the longer history of research on the subject is also included there (Resnick and Ford, 1981).

One of the most interesting efforts to combine cogitive science and the use of technology in the mathematics context is provided by the Buggy and DEBUGGY research and computer programs of Brown and Burton (e.g., Brown, 1978). Although this work was on addition and subtraction of integers, several of the objectives and byproducts of that research, and certainly its general philosophy, provide useful metaphors for work on higher level mathematics topics such as algebra and elementary functions.

More specifically, that work:

1. Approaches the analysis of algorthms in a procedural manner;
2. Assumes that there are systematic errors being made which, if identified, would be quite useful in helping the individual develop valid algorithms or strategies;
3. May be used to develop useful diagnostic support using computers which may increase the effectiveness of teachers in helping individuals identify their systematic algorithmic and conceptual difficulties nearer the time they occur.
4. Has interesting implications for strategies and materials to enhance the effectiveness of instruction and for teacher training. The research can be used, for example, to develop efficient lists of exercises designed to make the most serious conceptual and algorithmic difficulties "visible"; and
5. Can be used to raise specific and deep questions about how individuals process mathematical and other symbolic information.

Our long term plan is to take steps similar to these to develop computer-based diagnostic software sensitive enough to "expose" the most important conceptual and algorithmic misconceptions of students concerning topics in algebra and elementary functions. This plan is an initial step in exploring such questions as: what forms of representations do students have of these topics and are these representations "appropriate" ones?

The standard we have set is to diagnose or "profile" an individual's difficuties more reliably and efficiently than, say, a teacher can achieve with a relatively small class of students, unless s/he collects and evaluates homework problems on a very regular and thoughtful basis. Although the software is being developed and refined on a large computer system, we expect to redesign it for personal computers for use in pre-college settings. There it might be used as an "early warning system" to help indentify errant mathematical patterns nearer the time they are formed. The diagnostic profiles can then be used to direct the student to appropriate instructional materials.

Such an ambitious project would be impossible without resources such as sophisticated programming, design and technological support. But even more important is the capability to systematically examine and re-examine the performance of a large number of students on the many mathematical tasks of interest to us.

There is a risk, however, that the "trees and the forest" will be confused if attention is paid solely to the many interesting "bugs" or errors in algebraic technique. Sleeman has estimated the number of cases which must be examined by computer in his work with "malrules" invented by students in solving relatively simple linear equations and the numbers soon become astronomical (Sleeman, 1982). Such observations are absolutely critical but methodologies must be used which will ultimately help reveal deeper underlying patterns which cut across many topics--even though these patterns may be obscure to the student initially.

Our primary interest is in using our unusual opportunities for observation to attempt to determine or define the "cognitive objectives" (Greeno, 1976) appropriate to topics in precalculus.

The diagnostic process must ultimately be based upon carefully chosen categories of concepts, procedures and heuristics. These, in turn, might be quite useful in attempts to develop initial designs of semantic networks for the most important concepts to be learned. As Resnick (Resnick, 1981) points out, to attempt to define cognitive objectives effectively it is "necessary to combine conceptions of the subject-matter structure, general standards of well-structured knowledge, and empirically derived information about the knowledge structures of people who are recognized as expert in a domain".

One example of a higher level means of organizing observations of patterns is familiar to anyone who has taught algebra. It has been studied in some detail by Matz (1980). She examines the ubiquitous "linearity errors" $f(a+b) \text{-->} f(a)+f(b)$ which occur in settings involving many types of functions or algebraic expressions; for example,

$$(a+b)^n \text{ --> } a^n+b^n \quad \text{or}$$

$$\log(a+b) \text{ --> } \log(a)+\log(b).$$

The linearity class can be used to illustrate another important phenomena with interesting information processing overtones.

Much of algebra is very "procedural". Some problems require students to hold in memory or access many more procedures than others. Can some of the more persistent difficulties be explained in this way? A specific example is provided by the student who can be helped to do problems like expanding $(2x+3)^2$ without making the linearity error

$$(2x+3)^2 \text{--> } 4x^2+9$$

only to make precisely that error in

$$(x^{1/2}+x^{-1/2})^2 \text{--> } x+(1/x)$$

after studying procedures for dealing with fractional exponents.

Another illustration of this possible relationship between the number of procedures and performance can be given using equations involving rational algebraic expressions. Students can often master the separate procedures of factoring polynomials; finding least common multiples of denominators of several rational expressions; solve polynomial equations which result after "denominators are cleared"; but fail to do all these things in solving a given equation involving addition of rational expressions.

Other examples of tasks which should influence the definition of cognitive objectives might include: notation conventions (e.g., $2x^3 > (2x)^3$); parsing or chunking tasks (e.g., representing a complex algebraic expression with a single symbol); ubiquitous concepts such as equation, variable, and function; procedures for carrying out a specific type of algebraic task (e.g., creating or identifying equivalent forms of quadratic inequalities; or solving linear equations); and strategies and heuristics for approaching problems.

An important byproduct of the Center's work in these areas is the identification of more powerful heuristics which might be used to help reduce the sizes of the search spaces encountered as scholars of artificial intelligence attempt to design "intelligent" algebra programs for computers (Davis & Lenat, 1982).

Certainly the complexity of the topics treated in algebra and precalculus is significantly greater than in arithmetic. This prevents the comprehensive approach used in DEBUGGY. But what is lost there may be more than made up for in the more robust information available through careful examination of individuals' behavior with the higher level topics. Careful observation of large numbers of students on a large number of algebraic tasks provides a kind of longitudinal profile of their levels of understanding of concepts and algorithms, including arithmetic ones, which promises to be very useful--for example, in precollege instuction.

There is another important reason why we have elected, initially, to examine performance on a broad range of topics. The three facets listed by Resnick which must be carefully combined in the development of cognitive objectives are linked in ways which are of great importance in mathematics and which have serious implications for instruction. An example can be provided using the choice of algorithms in middle school for adding fractions with unequal denominators and the choice of algorithms taught the same student several years later (in second year algebra) for adding rational expressions.

If additon of fractions is taught in a certain way (e.g., obtaining least common denominators by applying the Fundamental Theorem of Arithmetic to the individual denominators, etc.), then the most powerful approach to adding rational expressions can be taught simply as a parody of it. Put differently, the choice of the mental represtentation students are encouraged to develop in middle school may have very important implications for their ability to pass to the next level of understanding or abstraction for that concept several years later.

Another example of the implications of the choice of representation at one level for the student's ability to build upon it at another level, perhaps years later, can be given using fractions also. In an informal paper (Research on Teaching Mathematics, 1980), the Research for Better Schools project of the National Institute for Education summarizes research on teaching mathematics. Three methods of dividing numerical fractions are discussed. The usefulness of some of these algorithms as a base for generalizations later in algebra varies greatly. One of the algorithms is:

$$\frac{1}{2} \div \frac{1}{3} = \frac{3}{6} \div \frac{2}{6} = \frac{3 \div 2}{6 \div 6} = \frac{3 \div 2}{1} = \frac{3}{2}.$$

This approach to the division of numerical fractions would not provide an effective way to divide rational expressions in algebra. One also wonders what fragments of it might be retained which would increase an already powerful tendency toward a very common error pattern. Passing from the second step to the third certainly seems to tempt a naieve student to believe that

$$\frac{a}{b} \div \frac{c}{d} \dashrightarrow \frac{a \div c}{b \div d} \quad \text{is a valid}$$

transformation. The research sited (Driscoll, 1979) indicates that, while all three approaches presented there seem to be equally effective on immediate testing, the approach given here seems more effective on retention tests. But it may also be achieving short term results at very significant long term cost in terms of the student's ability to build representations which facilitate generalizations to more abstract topics in the future.

These examples illustrate both the difficulty and the importance of determining the cognitive objectives and patterns in algebraic behavior that one should be attempting to evaluate. One cannot conceivably "drag" an individual through a battery of questions designed to profile his or her performance on all the basic techniques on which students normally drill in the typical algebra class. In fact, to do so obscures what algebra and precalculus should be about. How does one define, functionally, such procedures as "parsing or chunking" within algebraic expressions? What are appropriate stages in the development of students' "understanding" of the concept of "function" or "inverse function" and how can they be aided in achieving the most powerful mental representations of such concepts? These are among the many questions being formulated as we attempt to identify the most essential modes of "algebraic thought" to serve both as the objectives for instruction and for the computer-based diagnostic process to be used to help students.

_Activities underway in the Center_:

Work is already underway using research referenced here to carry out steps similar to the five listed above as characteristics of the DEBUGGY metaphor. More specifically, we are examining students' performance on approximately 1100 multiple choice problems given on precalculus mathematics examinations during the last three years. Approximately 3300 students have taken these tests which cover essentially topics each year and have been taught by approximately one dozen different faculty members. (Approximately 700 of these problems and their solutions are also on the University's computer system.) We are attempting to determine the most important forms of conceptual and algorithmic difficulty--especially those which persist even additional after instruction. Since the examinations are taken (out of class) by all students enrolled in a given semester, we have data on the performance of several hundred students on each of the items.

After identifying patterns of behavior on a topic which appear to be common across several instructors and large numbers of students, we develop prototype questions which will be used on Diagnostics which are _not_ multiple choice in design. Diagnostics are given weekly and currently consist of approximately ten problems or questions. Selected conceptual and error strands run through several diagnostics (e.g., the linearity types) so that we may monitor individual students' behavior on them throughout the semester.

The Diagnostics are worked by students using formatted paper and a carbon copy of their work is taken before they sign-on to the computer. There they may enter their answers and receive messages which are contingent upon the strategy they appear to have used in solving the problem. These messages identify the difficulty we believe the student has and analyze it. Many of these messages will soon be interactive. We will often take the student through a short tutorial so that they must apply the information just provided them.

The software also permits students to enter algebraic expressions of significant complexity as their answers and judges them.

The copies of the student's written work which are collected before they enter their answers on the computer are carefully analyzed in order to:

a) Determine all strategies (incorrect and correct) which students used in attempting to solve the problem;
b) Determine all answers they arrived at and their logic for each;
c) Determine what follow-up questions or problems might be used to effectively and efficiently reduce or eliminate inaccurate diagnoses of each student's strategy. The software currently exists to support asking additional questions in a wide variety of modes;
d) Test hypotheses about "linked" or repeated forms of errors or difficulties. For example, we can scan all questions of all Diagnostics which were designed to trap "linearity errors" (or other difficulties designed into the problems) across a variety of function types to determine whether tutorial or other instructional intervention is or is not effective. We can also remind students when they are continuing to make the same error but in different contexts. We hope that we can help students to "understand" that the linearity errors described earlier and many similar forms are, in a mathematically important sense, the same.
e) Determine strands of topics and concepts which can be used in the development of efficient tailored diagnostics in order to obtain a sufficiently complete profile to assign students to appropriate instructional resources (on- and off-line); and
f) Help determine means by which we can collect protocols on-line so that significantly less time is required to determine students' behavior on new problems than is currently devoted to the analysis of their written work.

The Diagnostics appear to be extremely effective in helping students to identify difficulties as they are learning or re-learning topics. They currently focus essentially upon algebraic technique. But this is being changed as we learn to formulate more suitable objectives and become more creative at thinking up tasks to determine whether they are met. For example, in addition to asking students to solve an inequality such as

$$\frac{2x - 1}{x + 1} > 0,$$

we are now planning to assign problems such as:

Assuming that A and B represent any algebraic expressions (and excluding values of the variable where the expression B might be zero when in the denominator), determine which of the following inequalities are equivalent:

$(A/B)>0$ $(A/B)>C$ $A\times B>0$

$A^2\times B>0$ $A^4 n<0$ $-1>0$.

One of the most powerful capabilities which the Center has as a result of these on-line materials is the ability to select a form of mathematical behavior on a topic of interest and, in a short time, identify a large subset of the several hundred students in precalculus who exhibit that behavior. We can then conduct interviews with them or use other protocols to obtain significantly more information concerning their misconceptions of the given topic.

## Conclusion:

The final facet of the Buggy metaphor concerned the desire to use the research summarized here to begin to develop testable hypotheses about how individuals process the kind of information encountered when learning algebra and elementary functions. Our observations are already hinting at some interesting phenomena but it is too early to describe them in detail.

An extremely important issue is, of course, the meaning of the word "understanding" in mathematics learning. Mathematicians are especially respectful of that word having normally studied topics in mathematical analysis (e.g., concepts such as "continuity" or "limit") four or five times beginning with an introduction to calculus in high school or college; and continuing through courses in advanced calculus; real and complex analysis; functional analysis and one or more courses in topology at the graduate level--each time believing that he or she has finally "got it"--only to discover additional levels of "understanding" of the most "elementary" concepts. The importance of these "levels" (and whether this is even a suitable paradigm) and of the ultimate "robustness" of the representations urged upon students cannot be overemphasized.

## Bibliography:

Brown, J.S. & Burton, R.B. Diagnostic models for procedural bugs in basic mathematical skills. Cognitive Science, 1978, 2, 155-192.

Carry, L.R., Lewis, C. & Bernard, J.F. Psychology of equation solving: An information processing study. Dept. of Curriculum & Instruction, University of Texas at Austin, Austin, Texas: 1978.

Davis, R.B. Some new directions for research in mathematics education: Cognitive science conceptualizations. Paper presented in the Begle Symposium at the Fourth International Congress on Mathematics Education, Berkeley, California, 1980.

Davis, R., Jockusch, E. & McKnight, C. Cognitive processes in learning algebra, Journal of Children's Mathematical Behaviour, 1978, 2, 1.

Davis, Randall & Lenat, D.B. *Knowledge-Based Systems in Artificial Intelligence*. New York: McGraw-Hill, 1982.

Driscoll, M. *Research within reach: Elementary school mathematics*. St. Louis, Missouri: CEMREL, 1979-80.

Greeno, J.G. Cognitive objectives of instruction: Theory of knowledge for solving problems and answering questions. In D. Klahr (Ed.), *Cognition and instruction*. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1976.

Greeno, J.G. Understanding and procedural knowledge in mathematics education. Educational Psychologist, 1978, 12(3), 262-283.

Kieran, C. Constructing meaning for the concept of equation. Master's thesis, Concordia University, Montreal, Quebec, 1979.

Matz, M. Towards a computational model of algebraic competence. Master of Science Thesis, M. I. T., June 1980.

Matz, M. Towards a process model of high school algebra errors. In D.H. Sleeman & J.S. Brown (Eds.) *Intelligent Tutoring Systems*. London: Academic Press, 1981.

Newell, A. & Simon, H.A. *Human Problem Solving*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1972.

Rachlin, S.L. Processes used by college students in understanding basic algebra Doctoral dissertation, University of Georgia, 1981.

Research on teaching mathematics: Summary of key instructional behaviors. Mimeographed, 12/11/80. Research for Better Schools, 444 N. 3rd Street, Philadelphia, PA 19123.

Resnick, L. & Ford. *Psychology of Mathematics for Instruction*. Hillsdale, N.J.: Lawrence Earlbaum Associates, 1981.

Sleeman, D.H. & Brown, J.S. *Intelligent Tutoring Systems*. London: Academic Press, 1981.

Sleeman, D.H. A rule-directed modelling system. To appear in LEARNING, 1982.

VanLehn, K. Bugs are not Enough: Empirical Studies of Bugs, Impasses and Repairs in Procedural Skills (Cognitive and Instructional Sciences Series Report CIS-11). Palo Alto: XEROX Palo Alto Research Center, 1981.

Wagner, S. Conservation of equation, conservation of function, and their relationship to formal operational thinking. Doctoral dissertation, New York University, 1977. *Dissertation Abstracts International*, 1978, 38, 5975a.

# But I Don't Want to Know All About ...

J. Arthur Seebach St. Olaf College

## ABSTRACT

As we at St. Olaf developed a flexible and integrated system consisting of a PDP 11/70 and a number of microcomputers, we discovered many things--many of which we didn't originally think we were interested in. I present here, in anecdotal form, some of the questions we have learned to ask and some of the solutions we have found. An isolated computer system is usually supported by its source to do whatever its developer intended it to do. Nobody really provides either the questions or the answers necessary for crossbreeding apples and oranges or for doing what *you* want to do. We have learned something about making our microcomputers and our PDP 11/70 work together, sharing programs and storage. We have also learned how to make large characters for classroom display on a 24x80 character screen and how to make an English speaking screen develop a French accent (for less than $50). Nobody told us how to do it; we asked questions.

## INTRODUCTION

This is meant to be a story; but a story with a point to it. Although the point of the story concerns both educational philosophy and technical issues in computing, there are essentially no references. That is part of the point of the story. When the development of computer science began at St. Olaf we were lucky enough to hire a very strong willed person to create a program which emphasized the intellectual core of computer science. He shamelessly brainwashed the lot of us in the Mathematics Department to the point where most of us think trees are really great and none of us can program in Cobol.

The stories that follow outline the pilgrimage from ignorance to moderate enlightenment which was necessary when we responded to a direct confrontation with microcomputers. This confrontation was to a significant degree sponsored by an NSF CAUSE grant designed to enhance the role of computing and data analysis in our mathematics and science curriculum. In particular, the author was provided released time for the amorphous area of microcomputer development. Since this is a formal paper, it should conceal much of the confusion and fumbling that was the hallmark of our progress. Faithful representations of chaos are difficult to follow.

We really did make plans and carry them out. Indeed, there was and still is a very strong sense of unity and purpose. The constraints of personnel and budget within which we worked meant that our approach was truly appropriate to a liberal arts institution in both facilities and style. General backgrounds and a willingness to learn replaced techincal expertise. Ph.D.'s in topology and complex function theory did not, ten years ago, require the sort of knowledge we found ourselves using.

Our goals were really positive, but frequently ended up being stated negatively. We wanted to avoid an awful lot of examples which we felt were bad. We thought that the UNIX operating system was great, but we were suspicious of the value of programming in BASIC. We wanted flexibility yet coordination, low cost yet significant power. Finally we wanted to do things our own way, while maintaining the ability to take advantage of outside work; we might prefer a new design but we didn't want to reinvent the wheel.

Flexibility required a stand-alone

microcomputer with its own disk drives and programming; coordination required communication electronics which could join the microcomputers to the main PDP 11/70, together with a reasonably standard screen display and keyboard. Low cost required some mass-marketed, relatively popular microcomputer; significant power required substantial memory as well as speed and a fast CPU like the Z80. Our wish for independence required an almost unheard-of level of access to the machine and its operating system; yet we needed access to a large base of commercial programming.

In June of 1980 we finally stopped looking for the ultimate machine (which didn't exist then, and still doesn't, so far as I can tell) and decided to accept the Heath/Zenith "All in One" (H89) as a good compromise. The H89 is not "the official" machine, but just the one that has received some institutional support and a lot of use. The H89 uses both Heath's own operating system HDOS, and CP/M. This is not an ideal solution to the problem of access both to details of the operating system and to a large base of commercial programming but it is a workable compromise. Full HDOS source code is available and CP/M is well supported with applications programming.

Mostly we have been lucky, but our learning has just begun. The individual "stories" which follow represent turning points on the way to where we are now. All the stories are true, but they have been ordered and sometimes organized for coherence. There were no textbooks for us to follow. We were neither a big university with a large and omniscient technical staff, nor a microcomputer users' group having fun.

## THE CENTRAL PROBLEM: COMMUNICATIONS

The first story (call it a case study if you wish) concerns the project which we felt was most basic: a link between the microcomputers and the PDP 11/70. Nobody around knew 8080 or Z80 assembler, nor anything about the Heath operating system. (CP/M would have presented the same problems but they came later.) The microcomputer had twenty-five pins on a little plug in the back that were supposed to be able to communicate with the outside world, but the plug in the wall to the PDP 11/70 had just four pins. Our physics technician knew a lot of useful things. Unfortunately neither he nor the rest of us had a vocabulary which allowed us to get from assembly language through the twenty-five pins to the four pins in the wall.

We started by learning about programming communications chips and learning a lot of fancy names for the signals which the various pins were supposed to provide. We learned that in sometimes means out: the phrase "signal in" may mean "into the micro" or "out of the micro and into the cable" to the device you are trying to talk to. Several sessions into the wee small hours and beyond generated a tremendous amount of respect for those who could deal with such arcane matters--until I found that one communications board on a computer whose manufacturer shall remain nameless had the wrong quartz crystal on it. All the baud rates were half of what I thought they were. All respect for the cleverness of those who programmed communications chips vanished. After a night of sleep I woke up with a lot of usable fancy new vocabulary, and names for many of those twenty-five pins. We all rapidly learned such acronyms as DTR, CTS, and words like "divisor latch". We discovered that our program would transmit ten thousand characters a second, so we learned how to make it wait until the 1200 baud line was ready. Now a "g" typed on the keyboard turned into a "g" somewhere else, just as the manuals said it should. We put the proper bytes in the proper registers and the communications chip sent signals just the way we wanted it to. Next we connected the microcomputer to the PDP 11/70 with just four signals--two grounds and signals in and out--and we seemed to run just like a dumb terminal.

Everything looked great until I made a dumb mistake and got the PDP 11/70 into an infinite loop. I knew that the way out was to type a control-C, so I did. Nothing happened! Not being willing to admit to the professional staff down the hall that we were in trouble, I tried the desperate measure of a control-D. That worked.

To make a long story short, we found that the operating system was grabbing the control-C but not the control-D. What was causing that? Our terminal program simply wouldn't transmit a control-C. Some operating systems intercept a control-C, others don't. The terminal was not yet dumb enough. It can take considerable effort to find out just what a particular version of a particular operating system does. The magic word "interrupt" passed before my eyes: some operating systems know about interrupts, and some don't. More experiments and finally success, the Heathkit really was a dumb terminal. Disabling interrupts allowed anything to pass through the microcomputer. Then we observed that the disk drive never quit running. We had killed the interrupt driven timing. So we learned how to disable only the interrupt generated by the keyboard. We were beginning to understand the operating system and the transfer of data inside the machine. You still can't read about such very important details for many operating systems. Source code from Heath helped once we learned how to read it. Today we can pass files back and forth over any terminal line. Plug in, run the terminal program, and follow the prompts on the screen.

As later stories will demonstrate, our development started with a recurrent theme: communications. One author of a technical book about microcomputers put it very well: A single manufacturer will sell you a lovely coherent system which you can happily plug together and run. But (to the tune of Professor Henry Higgens' refrain "Let A Woman in Your Life" from "My Fair Lady") put a plug into a port and your equanimity is gone; almost surely nothing will happen. The ideas behind RS232C communications ports and their signals are really quite straightforward, even if mildly complex. But the documentation provided by the average manufacturer is appalling. Two distinct manufacturers will provide documentation with even less communication than their products. If you want to develop a nice coordinated system of computing without spending ten to twenty thousand dollars per station, you will need to learn about the sorts of signals various devices are going to expect from one another. In particular you will need to learn how to read the sections of the manufacturer's technical manual on communications protocols and interface signals. While struggling though this, remember that Ph.D.'s in mathematics, the sciences and education are much less expensive than B.A.'s who just know how to solder.

If you want a nice system that does what you want it to do, it's likely to be all up to you. Most of the commercially developed systems I have seen were clearly conceived and designed by somebody in sales--probably a former history major who flunked junior high algebra. If you've got a Cray-1 idea and a Sinclair budget you are going to have to do it yourself--and learn how to compromise in the process. Much of what I read, Seymour Papert's "Mindstorms" among others, makes me think that we are going to have to become more active in doing worthwhile things or microcomputers are going to go the way of the new math, just not fast enough.

## KNOW YOUR OPERATING SYSTEM

Not long after we polished off the details of talking to the PDP 11/70 and passing files back and forth the now-famous "disk structure corrupt--please call Heath technical maintenance" started appearing on screens. But Heath technical maintenance couldn't help a student at midnight, and the student seemed concerned that his religion professor might not think well of "corruption" as an excuse for not handing in his paper on time. Here our investigations of the Heath Disk Operating System (HDOS) came in handy. With the help of the complete source listings for the operating system, just like the professionals with their source code for UNIX, we were able to use special utilities provided by the Heath Users Group to patch directory information and retrieve accidentally deleted and corrupted files. Nirvana! This even generated academically significant projects for systems theory students. The linked lists and pointer structures we discovered are important concepts which the students, and old line mathematicians for that matter, could see at work in the real world. And what an ego trip, to be in charge of your own (electronic) destiny.

Time and time again we discovered that the complete information which Heath had published about the operating system was very useful. With CP/M we had to spend considerable time gaining the same necessary information; much of what we needed to know does not seem to have been published. Raw access to the disk on a byte by byte basis is required to gain file compatibility between UNIX and the microcomputer operating systems.

For example UNIX ends each line with a single line-feed character and so does HDOS. Hence a file passed directly from HDOS to UNIX will work just as well in the framework of UNIX as it did in HDOS. However CP/M uses both a carriage-return and a line-feed at the end of each line. UNIX can't understand this, so everything ends up with an extra blank line for each line of text. On the other hand HDOS pads the ends of files with nulls, and this causes printer drivers in UNIX to have nervous breakdowns. The control-z characters which pad CP/M files don't seem to cause indigestion in UNIX. Everything works fine so far, but there isn't any way of being sure.

If more than one editor or formatter at opposite ends of the wire is to work on a file, you will have to know a lot of things that the salesmen can't even pronounce. You need to learn about line terminators, high order bits being set, and imbedded control characters, including tabs, lest each editor reject the other's work. The alternative is isolation, where the microcomputers live in a second-class world by themselves: UNIX has all the powerful stuff and you can't get at it except by turning your micro into a dumb terminal. Sharing requires knowledge. Our intention is to make the linkage transparent to the average user. All the necessary translations should be done by the routines which pass files; we needed to know exactly which translations had to be done and how to do them safely.

## EVEN HARDWARE COUNTS

Having gained confidence that we really knew how to make the system work for us, and that we knew something of the internal coding, we turned to graphics development. A Lear-Siegler terminal with a Retrographics board runs about $2000 and is just a dumb terminal with medium resolution graphics (approx. 512 by 256 dots). A study of the Heath circut diagrams whetted our appetites; there were plenty of dots on the screen and we knew in the abstract how to address each one. It was a matter of base two arithemetic. Then, Oh lucky day!, there was an advertisement saying that somebody had already solved all the messy electronic problems. Well, most of them. Excel in England had produced a board that would give Retrographics quality graphics on our microcomputers. By now we were nervous. Would the necessary information to make it work be available? We persuaded the U.S. vendor to send us a copy of the technical manual before we spent our limited grant funds on their product. The manual was well done; the coding was clear and the installation and operating procedures were well presented. We made the plunge, bought, and installed.

Nothing happened!! Well, a little fuzz showed on the screen. Phone calls, frustration, and more poking about the circut diagrams. We found it. The installation procedures described in the manual applied to a different version of the operating system and gave the one configuration in which the board would not work; we were trying to graph the operating system. The addresses chosen by the system for itself exactly overlaid the memory addresses used by the graphics memory. So we added memory, made a quick alteration to the address where the operating system resided, and the adaption to HDOS with its system-managed device drivers became an interesting student systems theory project.

By now we knew what we needed to know about the details of where the graphics memory really was, and how the operating system controlled access to this memory. Now for not too much more money we can have a microcomputer that can do its own thing or emulate a Tektronix scope. Still more time was spent staring at circuit diagrams and now we can do the same things with the H19 terminal for significantly less money.

The moral of this study is that things are changing so fast that no one really knows what's going on out there. Nothing quite works the way it is supposed to. Unless you are going to depend on very expensive "professionals" or just give up, you're going to have to learn some things. It can be good for showing your students how clever and adaptable (liberal arts) college faculty are. Remember those funny cloaks that drivers of early cars wore for protection against the demands of their none-too-reliable vehicles? Well, put on your intellectual dusters or keep driving horses.

Ah, yes! The story of the black and white color monitors. For a number of purposes we wanted large high-quality color monitors for the classrooms and an art department project. (The development of serious computer graphics for our art department, within a small college budget, is a technical morass all by itself.) We required good enough quality for students in the fifth row to be able to read characters; as well as the ability to use videotapes and other graphics output. All the characters we had seen were small, blurry, or expensive. (Apple's large characters came close, though a little blurry, but without expensive modifications the machines were hopeless for almost everything else we wanted to do.)

We had finally licked most of our problems except for the cost of good monitors. We had designed our own video output boards so we could use terminals with standard monitors and we had designed large character sets using the Excel graphics boards. Now we are a small college and not too proud to look for nonstandard ways out of a dilemma. We found an ad for used equipment that looked like just what we needed within the bounds set by our budgets. We called. Yes, these Conrac monitors had RGB color inputs. Yes, they had NTSC composite video inputs as well. Now you may know about as much about those acronyms as we did. If so you would also be in for a rude surprise. The NTSC composite video input did not have a color decoder! These monitors would take the NTSC color signal from a video recorder and show it in living black and white by careful projection though three color guns! That wasn't false advertising, it was just our lack of understanding of every last little detail. (We now know how to fix the monitors but just now we're too busy.) Our monitors have color tubes and crystal-clear RGB performance but just not with standard video recorders.

What is the message this time? Know the details of the so-called industry standards. All those impressive letters you see mean something. ( I have found technical people who have written technical material who when asked what does such and such mean really don't know; but my it sounded good.) Did you know that many popular microcomputers simply can't be used as terminals with various common minicomputer operating systems? Many microcomputer keyboards do not offer a full set of standard ASCII characters from control-@ through delete. If your keyboard won't produce 128 different codes, you're dead in a number of systems.

NONE-OF-THE-ABOVE PROBLEMS

For a number of years the telegraphic reviews for the American Mathematical Monthly have been prepared at St. Olaf. For reasons of efficiency we wished to put all this material into the PDP 11/70 word-processing system: we'd save retyping, manual sorting of material, and be able to do a very fast last minute printout. We solved the organizational problems and were all ready to give the system a trial run when we noticed that long lines of boldface, done by sending large numbers of escape sequences and ASCII characters to create slightly offset multiple strikes, were breaking in the middle and giving strange gibberish thereafter. What did length have to do with it? Why just on boldface lines?

By now we knew enough about counters and buffers to ask when, in terms of characters sent to the printer, was the break coming? Quick mental arithmetic in case after case yielded a number close to 512, a power of 2! After being told that such things couldn't happen, we finally persuaded a system programmer to look at the code for running the printer. Sure enough, somebody had built in a 512 character buffer which produced a line feed when it was full. So the many characters being sent to create one boldfaced letter were triggering the unwanted line feed. A quick change and the reviews suddenly became coherent.

This story concerns a special aspect of the questions which arise in communications. Devices must communicate through programs called device drivers which, for reasons of efficiency, usually keep blocks of characters in buffers in memory. When too many buffers start counting at once at different stages of the communications programming, an external sign of conflict usually appears. Random nonsense simply doesn't come out of computers even though buffer problems can produce a remarkable imitation; good luck in finding what's causing it. A knowledge of the general structure of communications programming is absolutely necessary. The whole problem is made more intricate by the great variety of interactions possible between multiple programs and hardware, and even more subtly, firmware.

SOFTWARE

Finally, even after you have learned all about hardware and interfacing and are fully prepared to run a program there is a catch 22. The software itself. Real information about how it will work is essentially unavailable. No manual I have seen does justice to what the program is really good for, nor to the special problems the program may have. But that is really the least of your worries. Most programs for microcomputers have been developed fairly quickly for a mass market. The developer must get his program out fast or lose an important share of the market. So there are bugs and there are updated versions with bells and whistles as problems and new demands are dealt with. On more than one occasion vendors have provided us with non-current or incomplete packages. So we had to discover the mistake and get after the vendors. (In all this I'm ignoring considerations of suitability of the software for its intended purpose.)

The question then is what is the latest version, and will it run on my machine? Considerations of which version of the machine you have are also important. How much memory do you have? Which version of the monitor program or the disk operating program do you have? Here many software vendors are clearly not successful in keeping up. The best you can do is hope to find a software dealer who is friendly and helpful, and who is willing to admit his mistakes. Make sure there is a mind and a real person behind the pinstriped gray suit. If the vendor is unable or unwilling to consider detailed technical questions, your problems will be multiplied. In general the people who wrote the program know pretty well what it will do. However, their programming ability has long since outstripped their command of plain English. You're on your own, like it or not.

There is a second aspect to the questions that need to be asked about software. What can you do about problems with software? The very interesting symbolic manipulation package muMath seems unable to factor some pretty simple things like the number 6. But this package comes with a language interpreter, and all the software is accessible source code in the form of text files. The manual even encourages using the language to modify the package (presumably after making a copy of the original which will still work). Alterations to most commercial packages are difficult or impossible. The only completely adequate documentation is the source code, though, of course, it's essential to have a nice precis called a user manual. The best of luck is if the two correspond. Usually they don't. In any case, be sure to investigate what you can do about flaws and just mere inconveniences in the programming. There are commercial compilers for languages that simply yield wrong arithemetic results. It's probably safe to say that somewhere out there even in UNIX there is a fatal flaw; we found a minor flaw just last year. The exact degree of software support by vendors ranges from hit-and-run to detailed help on a hot line with fully published source code. You have to choose your software with knowledge of what sort of help you can expect.

## CONCLUSION

Let us close this rich collection of embarassments by summarizing the questions we learned to ask and drawing some conclusions. The following list gives some of the things I believe someone is going to have to deal with when you get into a new system. I hope the order is logical; I know it isn't historical.

1. The operating system was designed to serve a general public according to a philosophy known only to the designer. You will have to find out exactly what that operating system does to your programs and files. In particular the details of I/O and related communications techniques must be known if you are not to end up with an isolated, clumsy, and educationally inappropriate system.
2. A second aspect of the operating system that needs to be understood is the disk operating system and file format. From the level of sync bytes to text file line terminators you will find that what you don't know will get in your way. You will need to know something of these standards and a great deal about the operation of your own system.
3. Yet another aspect of the operating system and its ability to communicate with you and the rest of the world is the area of device dirvers for printers, graphics and the like. There are standards in this area which are widely ignored in the personal computer market.
4. There are also industry standards for electronics, languages, and such technical details as keyboard construction and encoding. Knowledge of these will at least allow an efficient evaluation of the compatability of a new device, language, or program.
5. Software is probably the most chaotic world of all. Today very good and very very complex programs are available and have all the appeal of the almost marvellous. That's great. However, rapid development and a gee whiz appetite for bells and whistles ( I love them) will lead us all into flaky software problems. You will often be on your own. The manufacturers are too busy to take the time even in the cases where they care.

Now I'm not saying that even though ignorance is widespread nothing is being accomplished. Turnkey operations can do a lot of things. I am saying that as educators we should be leading; we must show the way. It is inappropriate as an educational example for us to build the philosophy of our educational uses of computing on the basis of products that are aimed at a mass market educated with a decades-old approach to very practical problems. If these machines aren't good for more than that, then we should all take up the study of art or music which are of much more human value. However, I don't believe that the situation is that bleak. For example, both the style and the content of the article on the UNIX operating system which appeared in the February 12, 1982 issue of Science give me reason to believe that there are possible approaches which are philosophically and educationally worth following.

Many of us tend to say "I'm a (fill in your profession); I don't want to know all about...". Indeed, this is not an entirely inappropriate sentiment. However I also don't believe that we can afford the cost of that point of view. The cost has two major aspects. The first and more subtle is educational and philosophical. There is a vast literature on education but most of it doesn't consider the issues raised here. Even though much of what is going on has all the negative benefits of brainless key pounding and games playing. The second cost is far more mundane, plain money. A willingness to learn and attack problems that just aren't professionally any of your business will, I am sure, allow us to implement 90% of a lot of very worthwhile things which we hear about at the research level but at costs between fifty and eighty percent (at least) below the "official" cost for doing it right.

In addition as educated educators we should be showing our colleagues and students what education is really good for--solving problems!

# ISSUES IN THE DEVELOPMENT OF COMPUTER GRAPHICS FOR MATHEMATICS INSTRUCTION

Mark J. Christensen

Georgia Institute of Technology

In this talk we shall describe the motivation behind the various key decision in our NSF CAUSE project at the School of Mathematics of the Georgia Institute of Technology.

Our goals were to: 1) Introduce information technology into the mathematics classroom and lecture hall as a demonstration tool. 2) Introduce and train faculty in the operation of the technology. In pursuit of the first goal, the development of materials was necessary. Originally we focused our software development efforts on the Calculus, but we have since become involved in other areas. The original materials were developed by the speaker, Les Karlovitz (Director of the School of Mathematics) and student programmers. All later materials were developed by other members of the staff, with technical assistance by the speaker and some programming assistance by students.

Fairly early it was decided that the Faculty should be directly involved in the development of software at all levels, including actual programming and coding. This decision was appropriate for a Mathematics Department at an institution such as Georgia Tech.

As our initial development had been done on the APPLE II microcomputer, we decided to retain it as our base machine. At the time of our earliest developments there were very few other machines available that fit our requirements of ease of use, high resolution graphics, cost and portability. For applications that require greater speed (typically floating point intensive ones) we use the TERAK microcomputer. An additional advantage of the TERAK is that it uses a larger (8 inch) disk drive.

For all later development (that is, beyond the Calculus), the language PASCAL was chosen. This choice was made for several reasons: 1) PASCAL is a high-level, block structured language. 2) It is reasonably transportable (in the UCSD form). 3) It was available on at least one other graphics microcomputer (the TERAK). Since the time of our decision UCSD PASCAL has become available on more machines, some of which also have graphics capabilities. Treating software and training as one-shot investments, PASCAL has tremendous advantages over BASIC.

In retrospect, we can add the following: 4) PASCAL is user-adaptable. That is, the user can write new library routines and insert them into the systems library. 5) Linkages between PASCAL programs and assembly language procedures (so often necessary in graphics applications) are accomplished in a very rational manner.

We should mention that features 4 and 5 enhance the transportability. Thus, it was fairly easy to develop a completely compatible set of graphics routines and install them in the library. Additional graphics extensions were added in this way.

A disadvantage of PASCAL is that the development cycle is longer. This occurs because PASCAL is a compiled language. Thus (especially on the APPLE II) there is a substantial amount of disk 'thrashing'. Trivial programming errors in a long program, that could be very easily corrected in an interpretative language, will take much longer to correct in PASCAL. Indeed, we often suggest that people try out ideas for algorithms (not complete programs) in BASIC first and then write the complete program in PASCAL.

Since we wanted our faculty to become proficient computer users a major effort was expended in running training seminars to introduce them to computers and to PASCAL. The response has been very good; much useful demonstration software in Probability, Statistics, Differential Equations (both ordinary and partial) has been produced. An important benefit has been that we now have a large percentage of the faculty who are computer-literate.

Once a piece of software has been prepared it is used in the classroom by the instructor. The students view the results of the computations on large classroom monitors. We decided very early in the project to utilize only black and white televisions. This decision was based on the following considerations: We could purchase high-resolution black and white monitors for approximately $500-700 in the 23 inch size class. Color monitors with the same quality of resolution and diagonal measure would cost between $2,000 and $3,500. We were very much afraid of purchasing color monitors that were 'good enough' for the color output of the APPLE II but would degrade the video signal produced by a computer with better signal generation circuits. In other words, we did not want to introduce a bottle-neck that would hurt us in a few years. Another important consideration was that the APPLE II did not generate a true NTSC color signal. This meant that even if we had purchased high-resolution color monitors we might not be able to see the color! Thus, resolution and fidelity of the display were the prime considerations for us. This policy has permitted us to use the TERAK computers in the classroom without loss of video quality.

In all this, it is our opinion, the limiting factor is neither the language nor the hardware. Rather, it is the difficulty in generating new ideas and viewpoints in the subject matter. Thus, it is clear that simulation will be useful in illustrating the laws of probability, but how to do so EFFECTIVELY? Likewise, it is clear that the development of the volume under a surface can be illustrated by the use of graphics, but how to do so EFFECTIVELY? Many ideas were tried out before (partially) satisfactory solutions were achieved. We feel that the direct involvement of the faculty permits this: the faculty know the limitations of the hardware and the language and thus naturally focus their efforts on the do-able.

# TEXT MATERIALS FOR COMPUTER LITERACY

Sarah Brooks

Mohawk Valley Community College

## Abstract

Text materials for computer literacy have been developed by Professor Robert Sanger and the author and are currently in use at MVCC in our Problem Analysis courses. The materials could be used in workshops for developing computer literacy in students and teachers. They cover the following topics:

1. The central processing unit
2. The fetch-execute cycle
3. Hycomp (a canned program which simulates a computer executing machine language programs, to illustrate 1 and 2 above)
4. Algorithms
5. Structure Charts (a combination of flowcharts and pseudocode particularly suited for designing structured programs)

# PRODUCTION SYSTEMS FOR COMPUTER BASED LEARNING

Alfred Bork

University of California, Irvine

[Note: This is a brief summary of Professor Bork's paper to be presented as part of the Panel on Software Production Systems. The full text is to appear elsewhere. This summary was prepared by the panel moderator.]

## Introduction

This paper is concerned with the variety of possible production systems for generating computer based and related learning material. A production system is the set of activities beginning with vague ideas about the concepts to be learned and ending with finished materials in the hands of users. These materials may include computer programs satisfying a variety of pedagogical needs, written material, visual material, and other learning aids. They may involve a variety of machinery for delivery to students--computers, audio units, videotape players, videodisc players, slide projectors, print material. I use the term "production system" rather than authoring system to emphasize that more than authors are involved.

The need to consider seriously production systems for computer based learning development is a result of many initial "cottage industry" efforts over the last twenty-five years. In the early days of developing computer based learning material little was known about how such material worked with students, and little was known about the production process. Not too surprisingly beginners in this new experimental medium began to produce material with a variety of strategies. At this early stage projects did not produce sizable amounts of material, because we still had much to learn about the capabilities of the new medium.

As we have moved toward the era of very inexpensive, personal computers, the possibility of large-scale production and distribution has become more and more viable. Little large-scale production and distribution is currently going on, but the prospects look very promising for the immediate future. Many companies are now considering the marketing of computer based learning material. Both school and home markets are important possibilities, stimulating large-scale development.

Hence, it is important at this stage to explore possible production systems. Without this step, sizable sums of money may be devoted to inappropriate approaches generating poor material. There are many affinities, often ignored, between producing curriculum material using the computer and producing curriculum material in older media, such as textbooks.

The production ideas which form the basis of this discussion were developed over a thirteen-year period at the Educational Technology Center. Thus, they are rooted in the experience of producing a very sizable body of curriculum material in a variety of academic areas for a wide variety of intended audiences.

## Criteria for Production Systems

It is reasonable, before discussing particular ways of organizing the production process for computer based learning material or intelligent videodisc learning material, to develop a set of criteria and emphases for such systems. That is, we need to ask what should be the properties of such systems before we begin to develop the systems directly.

The criteria to be discussed here include:

1. A major focus on pedagogical issues.
2. Use of the best possible material.
3. Encouragement of interaction as an active learning environment.
4. Encouragement of individualization.
5. Maximization of motivation.
6. Involvement of a wide range of good teachers.
7. Reasonable costs.
8. Estimable costs.
9. Ease of revision of the units.
10. Minimization of transferability problems.
11. Adaptability to large-scale production.

## Components or Activities of a Production System

Given the criteria suggested, we now consider the ingredients of possible production systems. Various production systems may all have the same fundamental activities. These components may be arranged in different ways; some may be omitted. Furthermore, the choice of who carries out each activity and how the developers at each stage interface with each other will differ from system to system. In some systems a particular component may be reported at several stages. The following components are discussed in some detail:

1. Preplanning activities.
2. Production management.
3. Objectives and specifications.
4. Training activities.
5. Pedagogical design.
6. Visual design.
7. Technical development.
8. Editing.
9. Evaluation.
10. Revision.
11. Marketing.
12. Distribution.

## Production Systems without Computers

These components are not peculiar to computer based learning material. Rather, they are stages that one might see in a process for producing any type of learning material--print, film, videodisc, or intelligent videodisc. A corollary to this is that the production process is not necessarily different for computer material than for other material. In many multimedia developments, a range of material, computer and others, may be produced. Hence, I now consider some existing systems for producing material in other media.

In looking at specific production systems for generating computer based learning material, it is reasonable to view systems for producing high quality educational material that have already been developed. Thus, we can hope to profit from the types of strategies already in existence. Development of learning materials has many common factors regardless of the media and the subject area, particularly if these materials are to be of the highest quality.

Many examples of past systems are available. We could examine the systems, for example, employed in the large curriculum projects in the United States in the 1960's, such as Project Physics. Or we could consider how such series as Sesame Place were produced. But we examine only two examples, textbook production and the Open University course production.

## Production Systems for Computer Material

### 1. Single individual systems.

In looking at production systems for computer material, I make a major distinction. First, I consider systems which were designed for single individuals (usually teachers) concerned with all the stages of the production process. These systems, although widely developed and widely advertised, are quite inadequate for the large-scale production problem at hand. They are capable, at best, of producing limited amounts of simple material. Experience has shown time and time again that single individual systems, under pressure of sizable amounts of development, move toward multiple individual or team systems. Any one individual can only produce a small amount, because that individual must do a variety of tasks. Strategies which allow the individual to concentrate on particular tasks, such as the pedagogical design task, are likely to lead to increased production. Furthermore, since most good teachers will not become good programmers, the supply of people who can write decent material with such systems will always be small, inappropriate for the curriculum material needed for full-scale use.

### 2. Team systems.

The first category of team production systems that occurs are those which represent the extensions of the single individual systems when serious production is considered. The single individual systems get into difficulty as the requirements increase. Then the developers of these systems begin to see that it is not necessary to have all the stages done entirely by one individual. So they begin to split the tasks, while maintaining essentially the same framework--languages and facilities. As might be expected, since the languages were not designed for this purpose, the results are unsatisfactory.

The major change from the single individual system in most situations is to split the pedagogical design stage and the technical development stage; but for each, the developers follow similar strategies to those of single user systems. Thus the "CAI language," or interactive capability, is used by a person other than the teacher, usually a coder. The teacher is then constrained to step 5 in the process.

An example of this modification of a single-user system was the development of the Control Data Plato authoring system after Control Data itself became heavily involved in Plato. While the system was at the University of Illinois, at least in the early stages, it was typically a single individual system, with one person doing everything. But when Control Data took it over, with needs for sizable production, it became a team system. The strategies for step 7, technical development, were essentially the same strategies that had been developed initially, but with people other than those responsible for pedagogical design. In this case the Tutor language, initially developed with a single individual strategy in mind, was extended to a more elaborate production system which, in most cases, separated the pedagogical design from the technical development. The programmers in step 7 use Tutor, although it was initially developed with teachers as the prospective users.

Beyond this extension of the one-user system, many variants are possible in a team production system. Each of the components already discussed allows a number of possibilities. The criteria stated earlier give a way of looking at any one such system. Much of the remainder of this paper is devoted to explicating one particular team-based production system, that developed in the Educational Technology Center at the University of California, Irvine. In the course of this exposition, I discuss some of the design considerations that led to the choice of strategies with each component of the production system.

## The Irvine Production System

The Irvine production system has been under development in the Educational Technology Center during the past 13 years. It has produced a very sizable amount of material, ranging from elementary school through university material and in a variety of modes. This approach has been well described in the literature, as have the resulting modules. I describe the system in terms of the way it typically operates and the way we would like it to operate if the resources were sufficient, using the production components described above:

1. Preplanning, e.g. negotiations with funding agencies, review of research literature.
2. Production management, based on Structural Analysis and Design Technique (Ref. 3).
3. Objectives and specifications, e.g. use of brainstorming sessions.
4. Training activities for teachers (pedagogical design), artists (visual design), students (programming).
5. Pedagogical design as a group activity.
6. Visual design by graphic artists using on-line design tools.
7. Technical development using structured coding.
8. Editing before and after coding.

9.-12. Evaluation, revision, marketing, distribution, considered together (briefly).

## Research Needed

At the present time developers of computer based learning material have extremely limited experience with full-scale production systems. We do not have the vast amount of experience that has gone into books, and even there many different variants of strategies have evolved. Hence, it is not too surprising that we have much to learn in this area and that research is badly needed about many of the details. Careful research in this area is critical for future progress in developing computer based learning material. I mention a few of the areas particularly in need of additional research aid:

1. Underlying learning capabilities.
2. Graphic and visual design.
3. Editorial control.
4. Pretesting before coding.
5. Variants of production systems.
6. Intelligent videodisc production.
7. Rapid coding technique.
8. Readability aids.

## References

1. Walter Perry, The Open University, (Jossey-Bass, San Francisco, 1977).

2. Alfred Bork, Learning with Computers, (Digital Press, 1981), Chapter 6.
3. Barry L. Kurtz and Alfred Bork, "An SADT$^{tm}$ Model for the Production of Computer Based Learning Material," Computers in Education, R. Lewis & D. Tagg (editors), North-Holland Publishing Co., IFIP, 1981.
4. Thomas W. Malone, What Makes Things Fun to Learn? (Xerox Palo Alto Research Center, 1980).
5. Tim Teitelbaum and Thomas Reps, "The Cornell Program Synthesizer: A Syntax-Directed Programming Environment," Communications of the ACM, September 1981, pages 563-573, Volume 24.
6. Alfred Bork, "Textual Taxonomy," (Technical paper, Educational Technology Center, January 1981).

# PANEL ON SOFTWARE PRODUCTION SYSTEMS

David A. Smith, Moderator

Duke University

This panel is jointly sponsored by SIG/MATH and SIG/EL-SEC-JC, and it has been organized by the SIG/MATH Program Planning Committee. It will be concerned with the entire process by which educational software is planned, developed, identified, recruited, tested, reviewed, and marketed.

The session will feature a major address by Professor Alfred Bork, Director of the Educational Technology Center at the University of California, Irvine, entitled "Production Systems for Computer Based Learning." (A summary of Professor Bork's paper follows. The full text will appear elsewhere.)

Professor Bork's lecture will be followed by a panel discussion that will include the following panelists: Dr. Harold Peters, Associate Director of CONDUIT (Iowa City, Iowa); Professor Michael Barnett, Brooklyn College, CUNY (Brooklyn, N. Y.); Denyse Forman, Research Associate, JEM Research (Victoria, B. C.); and Dr. Karl Zinn, Director of the Center for Research on Learning and Teaching, University of Michigan (Ann Arbor, Mich.). The lecture and commentary by the panelists is expected to be of benefit to authors and developers of software, those who would like to become authors or developers, and consumers seeking guidelines for assessing the quality of the finished product.

The committee responsible for organizing this session consists of David A. Smith, Duke University, Chairman; Michael Barnett, Brooklyn College; Sarah Brooks, Mohawk Valley Community College; Marjorie Fitting, San Jose State University; Sheldon Gordon, Suffolk County Community College; Ralph Heimer, Pennsylvania State University; Tony Jongejan, Everett High School, Everett, Washington; Marge Kosel, Minnesota Educational Computing Consortium; Theron Rockhill, SUNY College at Brockport; Ronald Wenger, University of Delaware; and Karl Zinn, University of Michigan.

# COMPUTER-ASSISTED DISCOVERY IN THE MATHEMATICS CLASSROOM

Marjorie A. Fitting - San Jose State University

Color, graphics, immediate feedback, and speed enable a computer to provide unique interactive learning experiences using CAI. Computers are efficiently used for drill and practice, and games can be used to motivate these activities. However, more creative uses can include tutorials, demonstrations, simulations and problem solving. Three of these, tutorials, demonstrations and problem solving, will be illustrated with slides from actual computer sequences. The methods used to develop mathematical concepts at various levels of high school mathematics and will be discussed.

1) Tutorials can allow the user to discover the principle to be learned and to practice the principle discovered.

2) Demonstrations of cause-effect relationships involve a single independent variable and immediate display of results. Two or more such relationships can be combined for multivariable results. In each case the user decides which values to enter and the computer creates and displays the results.

3) Computational procedures allow the gathering of data and operations on the data which assist the user in finding a solution to a particular problem.

# IMPACT OF COMPUTERS ON PRECOLLEGE MATHEMATICS:

# THE PRESENT AND THE FUTURE

Jerry Johnson (St. Olaf College, MN)

Tony Jongejan (Everett High School, WA)

Cheryl Lemke (Adna High School, WA)

Computers are rapidly being placed in high schools. Within the mathematics community, most of the current concerns are focused on the immediate impact of the computer's presence---that is, the "what-can-be-done-tomorrow" syndrome. What software is available? Which students should be exposed to the computer? Which teachers are interested in (or capable of) using the computer? Each of these questions will eventually fade in importance. All students will be exposed and all teachers will possibly use computers in their classrooms IF such exposure or usage can be shown to enhance or promote learning. Unfortunately, very little attention is being given to the full impact which computers will have (can have?) on the educational process, especially within the context of mathematics education. Somehow, a balance must be struck between the short-term concerns and the long-term concerns.

The session will attempt to look at some of these issues, with the focus being on a planned integration of computer technology into the mathematics classroom so as to ensure and maximize the positive effects on a long-term basis. The speakers will use their experiences to approach the issues from three perspectives: the small high school situation; the large high school situation; and, the coordination of high school content and activities with college expectations.

In a look at both the present realities and the possible futures, the session will explore the impact of the usage of computers (in any form) on the mathematics curriculum, the processes of learning mathematics, the changing goals for learning mathematics, and the roles of the mathematics teacher. Examples, research, and models will be discussed and critically examined. An orientation towards the future of mathematics education and the real possibility of having some control over that future will help reduce some of the chaos which currently exists.

# THE SWARTHMORE COMPUTER TRIGONOMETRY PROJECT

Eugene Klotz - Swarthmore College

We are recipients of a grant under the NSF-NIE Program for Mathematics Education Using Information Technology, and are developing a series of trigonometry tutorials. We are using Atari computers with a disk drive and 48K memory.

Our work is visually oriented, using color graphics as much as possible, minimizing text. The pace and order of presentation are under user control via joystick and the four special keys on the Atari keyboard. Upon power-up the user is given the option of an instruction set to make him/her comfortable with the joystick and the various features of the programs. Whenever possible, we use the joystick to elicit motor involvement on the part of the student.

To assure educational soundness and relevance, our units are being written by a team of three high school mathematics teachers. For motivation we use color (also important for clarity and for emphasis), sound (earphones can, of course, be used in quiet environments), and games (all with learning objectives). Our units are aimed at the mainstream high school trigonometry population as supplemental material. However, we expect the units to be useful to a much broader population.

We are using Atari computers because of their low price and many graphics features. These include a number of graphics modes with varying color capabilities and resolutions (highest: 190 × 320). In addition, there are several character modes which can be used for letters of different sizes and colors, and can be redefined by the user, giving further possibilities both for graphics and animation. Moreover, there is a flexible display list system which allows mixing many graphics and character modes on the same screen, and which also greatly increases color possibilities within the same mode. The "player-missile" graphics system provides several vertical strips within which one can place small objects--in colors independent from the screen colors--and then conveniently move the objects as desired. Fine scrolling is a very important hardware feature for us. It allows one to treat the screen as a window into a much larger world, by moving smoothly in any direction. This can afford a good solution to the very serious and universal program of how to display more information than conveniently fits on the screen at one time.

Our underlying programming is being done in the FORTH language. Although this applications language has been used for everything from pocket language calculators to radio telescopes to arcade games, it is little known in the academic community. The reasons for our choice were: speed, compactness, flexibility, total control of the computer, ease of programming, and availability. We are quite pleased with FORTH, but should point out that it is different from most standard programming languages. Fortunately, a splendid new book on FORTH has just been published (Starting FORTH, by Leo Brodie, Prentice-Hall).

We are testing some of our preliminary units in several high schools during the second semester of this academic year. By the end of this summer, we expect to have completed an Introductory Trigonometry Package, which will cover angle measure and the trigonometric functions. This package will be sent to MicroSIFT and to CONDUIT for comments, evaluation, and possible distribution.

Commercial vendors may also be approached. We hope the Introductory Package will be available to users by the second semester of academic year 1982-83.

We shall be looking for further funding in hopes of producing a package covering: preliminaries (special right triangles, pi, etc.), triangle trigonometry, trig identities, inverse functions, and polar coordinates. We are laying aside the work we have done in these areas in order to focus on the Introductory Package.

# MUSIC TERMINOLOGY OF CHILDREN, COLLEGE STUDENTS, AND FACULTY: APPLICATIONS FOR COMPUTER-ASSISTED LESSONS

Harriet I. Hair

University of Georgia

## ABSTRACT

The purpose of the present study was to determine (1) what vocabulary subjects would use when asked to write ten descriptors for a given music concept word, and (2) if consistent terminology would be given by children in grades four through seven, by college students, and by university faculty members. The music concept words given were: "loud, soft, fast, slow, high, low, timbre, (tone color) harmony, minor, and rhythm." Although there is more similarity in the vocabulary of adults and children of these ages than with younger children, there remain important differences in the vocabulary of these groups. The faculty chose the least similar vocabulary to that of the children. Yet, it is the faculty who are writing the computer lessons for children. Adults tended to give more synonymous and technical terms, while the children gave more purely descriptive vocabulary.

## STATEMENT OF PROBLEM

Authors of computer-assisted lessons for children need to have an understanding of the vocabulary that is meaningful for children. In developing teaching materials one should be aware that children have different vocabularies: written, spoken, visual, and aural. And, these are constantly changing depending upon the child's stage of cognitive development. These different vocabularies will influence the child's responses to the lessons. The present study was limited to the written responses given by children and adults to describe music concepts.

Many words used by adults to describe music are confusing to children. The accepted linguistic meaning of a word that a child learns initially, may not be the same as the musical meaning. For example, children learn the words "high" and "low" in relation to space. Only later do they learn to associate these terms with the pitches of sounds. The tactile term "soft" is used by adults to denote sounds of lesser volume. The musical term "minor" refers to the tonal organization of a commonly used scale.

Researchers have established that children can make aural discriminations among musical sounds long before they can use the appropriate music terminology for labeling these differences (Andrews and Deihl, 1967). Hitchcock (1942) suggested that sound-descriptive terms might be more appropriate for teaching young children than traditional music terminology.

Utley (1970) in developing his music perception test, used words already familiar to the children, not the language used by trained musicians. He asked junior high school students (grades seven through nine) to listen to music and to describe it in their own words, using "non-technical" terms. He found substantial agreement among the choice of vocabulary of the various groups and that the words used by one group were "readily understood" by the others. (p. 45)

Farnsworth, (1969, p. 80) in citing research by Schoen, Gatewood, Mull and others, stated that synonymous words will be used with some consistency to describe our Western music if the listeners are drawn from roughly the same subculture. He felt: "The degree of agreement is little affected by differences in listener intelligence, tested musical aptitude, musical training, or age level (if above the sixth grade)."

---

The author wishes to acknowledge the assistance of John S. Edwards in the computer design and programming.

In an earlier study, Hair (1981) found that when young children, aged seven, eight, and nine, and college students correctly labeled music concepts, they used consistent and traditional terminology. But when incorrect terms were given there was little similarity between the vocabulary of the two groups. The number of different words given per concept ranged from 23 to 72. Yet, only one to five words per item were common to both children and adults.

The purpose of the present study was to determine (1) what vocabulary subjects would use when asked to write ten descriptors for a given music concept word, and (2) if consistent terminology would be given by children in grades four through seven, by college students, and by university faculty members. The choice of grade levels in this study included children in lower grades and the same grades as those cited by Farnsworth and Utley. It included only the highest grade used in the earlier study by Hair.

SUBJECTS AND PROCEDURE

The 275 subjects in this study were composed of 110 children in grades four through seven, 141 college students, and 24 university faculty members. The subjects were given a printed form with the following instructions: "List the words you could use to describe each of the following musical concepts. Try to give 10 words per concept." The concept words given were: "loud, soft, fast, slow, high, low, timbre, (tone color) harmony, minor, and rhythm."

The data were coded for computer processing and analysis. Demographic information given for adults was sex, class standing, major, major concentration, applied instrument studied, and years of music study. For children, only grade level and gender was obtained. Presently, only computer programs for alphabetical listings of single words according to class standing have been done. Future research will involve additional analyses and descriptions of the computer programs.

RESULTS

The vocabulary for each music concept category was analysed according to: 1) the frequency of words given by each group, and 2) the words common to children and adults. A comparison of terms used most frequently per concept is given in Table 1.

For item 1 (loud), the term "noise" was ranked highest by the children, fairly high by college students and low among faculty members. The children's second choice "high" was only rarely found among adults. The two words ranked highest by adults were "forte" and "big." The technical term "forte" was not given by any children, and they ranked the term "big" very low. Adults frequently used the words "volume" and "forceful." Children did not use these terms.

An analysis of the words common to all groups showed that of the 33 words given by children, 32 were used by college students, and 18 by faculty.
See Table 2.

On item 2 (soft), the children's highest ranked word was "low." Their second choice was "quiet." "Low" was ranked sixth for college students, while only two faculty used this term. "Quiet" was the highest ranked word for both adult groups, followed by "piano" or "pianissimo." No children used the latter terms. The words "whisper" and "light" had high rankings for all three groups.

For words common to children and adults, of the 29 words chosen by the children, 28 were given by college students and 19 by faculty.

On item 3 (fast), the word "quick" was ranked first for both adult groups and second for children. "Speed" was the children's first choice, while this term was second choice for college students, and fourth for faculty. Adults ranked the term "rapid" next, while only three children gave this term. The word "run" rated highly among college students, but it was much lower for children and faculty. Adults used the term "allegro," but the children did not. The word "car" ranked third for the children.

There were 26 words common to children and adults. College students listed 25 of the children's words, and faculty, 17.

On item 4 (slow), the children chose the word "turtle" most often. This was not in the upper rankings for college students, and only one faculty member used this word. The term "walk" ranked fifth for children, and second for college students. No faculty used this word. The adult groups ranked the word "drag" second and third. Otherwise, college students most frequently gave technical terms "largo," "andante," "adagio," "tempo," and "ritard." Faculty used the words "relax," "easy," "crawl," and "deliberate." Additional frequent choices of the children were "fast," "soft," and "slow."

There were 33 words common to children and adults. College students used 31 of these and the faculty, 20.

On item 5 (high), the pitch related terms chosen most often by the children were "sky," "building," "flute," and "squeak." The word "loud" was ranked first, and "soft," fourth. The highest ranked word for both adult groups was "up." Only two children gave this term. Adults frequently chose the words "soprano," and "pitch." College students also gave the words "top," squeak," and "treble," while faculty chose "shrill," "bright," and "piercing." Except for the term "squeak," only one child, or none, used these words.

Of the 41 words common to all 3 groups, college students used 40 words and the faculty used 15.

For item 6 (low), both adult groups gave the highest three rankings to "bass," "deep," and "down." Children ranked "deep" second. The other two words were seventh and eighth. The term "ground" was fourth among the children's choices. Among the non-pitch related terms, "soft" was given most often by the children, with "loud," given as third. Only one child used the term "pitch." Adults, especially college students, chose this rather frequently.

Of the 27 words common to children and adults, college students listed 26, and faculty, 13.

On item 7 (timbre), nearly half of the children and one-fifth of the college students gave no response to this concept. Of the words given most frequently, the term "color" was ranked first for children, seventh for college students, and fourth for faculty. Note that the terms "tone color" were given on the printed form. A few children used references to "the way the music sounds" and "color you can hear." Adults often gave the terms "sound," "quality," "instrument," "dark," and "bright."

There were 31 words common to children and adults. Of these, college students used 30 words, and faculty, 18.

For item 8 (harmony), approximately one-fourth of the children left this item blank. Children chose the term "sound" most often. College students ranked this term second and the faculty, third. However, adults usually used this word in the context of simultaneous or multiple sounds. The rankings for the term "blend" were second for children and fourth for college students. Only three faculty chose this word. The highest ranked word for college students was "together," for the faculty, it was "chord." The latter term was in third place for college students. No children listed the term "chord."

Of the 26 common words for children and adults, college students used 24 words, and the faculty, 12.

For item 9 (minor), 37 percent of the children, 14 percent of the college students, and 12 percent of the faculty gave no response to this item. The words for the children's group ranked as follows: "soft," "major," "piano," "flat," and "loud." Two children stated "not major." Note also that two children wrote "instead of a flat tire, flat music." Most of the words chosen by children were given less than three times. The word "sad" ranked first for college students and third for faculty. Children did not use this descriptor. Other terms used frequently by college students were "not major," "scale," "sound," "flat," and "chord." The top ranked words for faculty were "chord," "key," "sad," "dark," "interval," and "not major."

There were 19 words common to children and adults. College students gave 16 words, and faculty, 11.

On item 10 (rhythm) 15 percent of the children left this item blank. The highest ranked term for all three groups was "beat." The terms "fast," "steady," "tempo," and "slow" ranked high for the children. They also gave terms such as "music," "sound," "loud," and "soft." Adults most frequently listed the terms "pattern," "pulse," "music," "time," (tempo) "steady," and "meter." Children did not use the terms "pulse" or "meter."

Of the 23 words common to children and adults, college students used 21 words, and faculty, 12.

## DISCUSSION

Although there is more similarity in the vocabulary of adults and children of these ages than with younger children, there remain important differences in the vocabulary of these groups. The number of words common to children and faculty was much smaller than between children and college students. Thus the faculty chose the least similar vocabulary to that of the children. Yet, it is the faculty who are writing the computer lessons for children.

Adults tended to give more synonymous and technical terms, while the children gave more purely descriptive vocabulary. The fact that many children and some adults did not give any responses to the more technical musical terms "timbre" and "minor" illustrates the lack of understanding of these terms.

The following recommendations should be considered in developing computer-assisted lessons for children. Music educators should: 1) analyse these differences in vocabulary between children and adults, 2) use vocabulary that the children understand, make paired associations with the new words being taught, and teach the musical meaning of the words, 3) be more aware that children do not understand the technical terms and that use of more descriptive vocabulary might help in establishing conceptual meanings, and 4) provide additional materials for teaching pitch related terms.

One of the primary goals of computer-assisted lessons in music would seem to be that the child could use these for independent study, enrichment or remedial purposes. Thus, it would appear crucial that the authors of these programs should use vocabulary that is meaningful for the child.

## REFERENCES

Andrews, F.M., & Deihl, N. Development of a technique for identifying elementary school children's musical concepts. Final Report BR-5-0233 (September, 1967), Pennsylvania State University, ERIC: ED 016 517.

Farnsworth, P.R. The social psychology of music. Ames, Iowa: The Iowa State University Press, 1969, 80.

Hair, H.I. Verbal identification of music concepts. Journal of Research in Music Education, 1981, 29, 11-21.

Hitchcock, A. The value of terminology in children's descriptions of changes in pitch direction. Unpublished master's thesis, University of Minnesota, 1942.

Utley, E. The development of a musical perception test free of technical vocabulary for use in grades six through twelve. Council of Research in Music Education Bulletin, 1973, 35, 44-52.

Table 1

Words Chosen Most Frequently By Group

| Item | Children N=110 | College Students N=141 | Faculty N=24 |
|---|---|---|---|
| LOUD | noise 28, high 24<br>drum 11, hard 9, boom 9<br>deep 7 | forte 67, big 33<br>strong 32, noise 24<br>volume 24, forceful 21 | big 9, forte 8, full 8<br>forceful 7, volume 7<br>sound, 7 |
| SOFT | low 31, quiet 19<br>loud 11, smooth 8<br>sound 7, whisper 6 | quiet 73, piano 62<br>lightly 33, whisper 28<br>gentle 19, low 17 | quiet 10, piano 9<br>whisper 6, easy 5<br>gentle 5, light 5 |
| FAST | speed 25, quick 15<br>car 8, fast 8<br>slow 7, high 6 | quick 75, speed 46<br>rapid 36, allegro 31<br>run 31, hurry 29 | quick 14, rapid 10<br>move 8, speed 7<br>swift 6, run 6 |
| SLOW | turtle 20, fast 14<br>soft 13, slow 8<br>walk 6, move 4 | largo 19, walk 16<br>drag 15, andante 14<br>adagio 13, tempo 13 | relax 7, drag 6<br>easy 5, crawl 4<br>deliberate 4 |
| HIGH | loud 28, sky 8<br>building 7, soft 6<br>flute 5, low 5 | up 46, soprano 31<br>pitch 23, top 17<br>squeak 16, treble 15 | up 9, shrill 6<br>bright 5, piercing 5<br>pitch 5, soprano 5 |
| LOW | soft 21, deep 19<br>loud 9, ground 8<br>sound 6, low 5 | bass 48, deep 46<br>down 35, bottom 24<br>pitch 18, sound 13 | deep 11, bass 6<br>down 6, under 6<br>base 5, pitch 4 |
| TIMBRE | color 8, music 6<br>loud 4, way 4 | sound 32, quality 28<br>dark 21, instrument 20<br>bright 17, different 14 | quality 10, sound 10<br>tone 7, color 6<br>dark 5, bright 4 |
| HARMONY | sound 11, blend 7<br>slow 7, soft 7<br>melody 5, minor 4 | together 36, sound 34<br>chord 30, blend 26<br>note 24, two 21 | chord 8, melody 4<br>sound 4 |
| MINOR | soft 8, major 6<br>piano 5, flat 4<br>loud 4 | sad 38, major 20<br>scale 19, sound 16<br>flat 15, chord 11 | chord 8, key 5<br>sad 5, dark 4<br>interval 4, major 4 |
| RHYTHM | beats 47, music 9<br>fast 7, steady 7<br>sound 6, tempo 5 | beat 109, pattern 26<br>pulse 23, music 18<br>time 18, steady 17 | beat 9, pattern 8<br>music 7, tempo 6<br>meter 5, pulse 5 |

Table 2

Words Common To Children and Adults

| Item | Words |
|---|---|
| LOUD | big 46, blast 8, bold 6, boom 14, car 2, crash 9, dark 2, deep 11, drum 14, ear 8, hard 24, harsh 13, high 34, horn 3, hurt 4, large 7, loud 3, music 4, noise 54, pitched 5, playing 3, pow 2, radio 2, ringing 2, rock 2, scream 16, soft 16, sound 28, strong 41, tone 3, truck 3, tuba 2, yell 10. |
| SOFT | easy 21, faint 3, far 3, feather 6, flute 4, fur 4, gentle 26, graceful 4, hard 4, hear 7, high 4, light 43, listen 3, little 14, loud 25, low 50, music 4, noise 6, peaceful 19, pleasant 7, quiet 97, sigh 2, silent 3, slow 11, smooth 18, sound 27, sweet 12, violin 5, whisper 40. |
| FAST | airplane 4, beat 12, car 9, cheetah 5, dance 3, fast 11, flow 2, funny 2, go 7, high 8, hurry 36, measure 4, move 38, note 8, quick 104, rabbit 4, race 12, rapid 49, rhythm 5, run 43, slow 14, speed 78, swift 13, tempo 24, up 7, zip 4. |
| SLOW | beat 10, boring 2, deep 2, drag 23, easy 14, fast 25, gentle 4, going 3, little 2, long 13, low 8, measure 4, motion 6, move 11, music 4, note 6, peaceful 5, quick 7, quiet 7, rhythm 4, sad 3, slow 12, sluggish 11, smooth 6, snail 10, soft 16, sound 4, speed 12, steady 9, tempo 16, turtle 29, walk 22, waltz 7. |
| HIGH | air 4, bell 3, big 2, bird 5, bright 12, building 8, clarinet 5, cloud 3, ear 6, fast 12, flute 10, high 15, hurt 4, loud 30, low 10, mountain 3, music 3, noise 4, note 9, piano 5, piercing 20, pitch 29, reach 8, sharp 13, shriek 2, sky 13, skyscraper 3, soft 8, sound 18, squeak 22, squeal 2, strong 3, sweet 3, tall 13, thin 4, tone 2, treble 16, tree 7, up 57, voice 10, whistle 2. |
| LOW | base 7, bass 58, big 3, deep 76, down 45, drum 4, easy 2, ground 11, heavy 11, high 8, little 2, loud 11, low 15, mean 3, music 3, pitch 23, quiet 5, rumble 7, short 8, slow 14, small 3, smooth 2, soft 28, sound 22, tone 3, tuba 9, voice 11. |
| TIMBRE | blend 5, blue 5, brass 6, brown 2, color 23, create 3, deep 4, different 11, fast 7, happy 3, harmony 5, hear 4, instrument 24, loud 12, mellow 9, mood 3, music 10, oboe 2, pretty 2, red 5, rhythm 3, sad 2, saxophone 2, smooth 6, soft 11, sound 44, texture 9, tone 14, trumpet 3, voice 9. |
| HARMONY | blend 36, chorus 2, feeling 2, go 5, high 5, lovely 2, low 5, major 3, melody 22, minor 5, music 15, note 28, part 10, peaceful 3, piano 3, played 7, pretty 2, rhythm 5, soft 9, song 5, sound 49, sweet 5, together 40, tone 12, tune 7, two 22. |
| MINOR | flat 20, form 2, half 9, important 3, key 15, la 2, little 4, low 10, major 30, minor 3, music 6, note 2, quiet 2, sharp 9, short 4, slow 4, small 8, sound 19, unimportant 2. |
| RHYTHM | beat 165, clap 6, different 4, drum 5, equal 3, fast 13, foot 2, good 2, kind 2, music 34, note 14, pattern 36, play 2, rhythm 3, slow 9, smooth 4, song 7, sound 14, speed 4, steady 24, tempo 23, vary 2, vibration 3. |

# THE USE OF A TABLE DRIVER FOR INDIVIDUALIZED DESIGN OF COMPUTER-BASED INSTRUCTION MATERIALS IN MUSIC THEORY

Michael Arenson

University of Delaware

## ABSTRACT

Within the last year, a table driver design has become the framework of the music theory lessons developed on the PLATO® system at the University of Delaware. In the lesson table driver, the instructor is given a skeletal framework within which to construct the lesson. Variables such as competency level can be set by the instructor so that the instructional needs of his/her student group can be met. It is hoped that this new design will allow the music theory lessons to be useful to a wide variety of instructors and students.

## BACKGROUND

In the spring of 1978, the music theory division of the Department of Music at the University of Delaware received an Improvement of Instruction grant from the Center for Teaching Effectiveness to develop a competency-based education (CBE) program in music theory to be used in conjunction with the core music theory courses in the Freshman and Sophomore years. The PLATO system was designated as the delivery system for the CBE program. Since that time, thirteen lessons have been developed for use by students enrolled in the first-semester music theory course for majors and a music fundamentals course for non-majors. Table 1 is a listing of these lessons.

All lessons are of the drill-and-practice variety. Tutorial lessons are destined for a later time. Lessons 1 - 7 were designed and developed during the period from 1978-80. The design of the lessons was accomplished by the music theory division. Each member of the division was responsible for the initial design prototype for certain lessons after which the entire group discussed each one and made modifications to the design. Each lesson design consisted of a layout of the basic terminal screen display, the competencies expected, and appropriate introduction and feedback messages to be received by the student for each of the units in the lesson.

Table 1

Music Theory CBE Lessons Developed at the University of Delaware

1. Pitch Identification
2. Octave Designation Identification
3. Half Steps - Whole Steps
4. Beat Units and Divisions
5. Meter and Time Signatures
6. Scales
7. Key Signatures
8. Written Intervals
9. Diatonic Chords
10. Transposition
11. Basic Part Writing
12. Bass Figurization
13. Partials

Following the design stage, the lessons were written on the PLATO system. Following student tryouts and lesson review by the Office of Computer-Based Instruction (OCBI) staff and certain content experts, lessons were considered ready for student use (1981). The lessons have worked well but they are rather inflexible. For example, in the scales lesson, the students are presently given the option of being drilled and tested on (1) major, (2) minor, (3) modal scales or (4) a mixture of all types of scales. The lesson was designed with those four options because that was what the music theory staff had thought was the most practical design and the one that would work well for most students in most educational situations. However, if an instructor decided, for example, that he/she wanted his/her students to drill only on the major, Lydian, and Mixolydian modes (modes that have almost the same intervallic sequence), then the internal code for the lesson would have to be changed to create a unit that would allow for a mixture of the three modes. Likewise, if the competency which is now set at 90% were considered too high, the programmer would have to go into the code and make an adjustment. In other words, the traditional design has not allowed for very much instructor control.

## A Table Driver Design

Within the last year, thanks to the expert programming of Mr. Paul Nelson, a student at the University of Delaware, lessons have been written that allow for instructor control of lesson design. Utilizing a table driver, instructors can actually design lessons to meet their own specific educational goals. For several years now the table driver design has been an integral part of the University of Delaware GUIDO lessons (1978) but has only recently become the framework of the music theory lessons. The lessons developed during the last year include lessons 8 - 13 of Table 1. Lessons 8 and 9 replace earlier lessons written in the traditional format.

In a table driver, the instructor is given a skeletal framework within which he/she constructs the lesson. The table drivers for all the lessons have some properties in common and others that are unique to each lesson. When the instructor enters the table driver for any one of the lessons 8 - 13 of Table 1, he/she receives a screen display similar to Table 2.

Table 2

Entry Screen Display in Music Theory Lesson Table Driver

Press NEXT for an explanation
SHIFT-NEXT to skip it
LAB to change the title
SHIFT-DATA to see/change the access list for this table driver
SHIFT-LAB to enter a purpose page

The keys "NEXT", "SHIFT-NEXT", "LAB", "SHIFT-DATA", and "SHIFT-LAB" are all found on any terminal utilizing the PLATO system. The programmer of a lesson designates how each key will function when pressed by the user. In the case of these music theory lessons, Mr. Nelson chose the designation for the keys as indicated in Table 2. By pressing "NEXT", the instructor is taught how to use the table driver. By pressing "LAB", the instructor creates the title for the lesson. "SHIFT-DATA" controls access to the table driver through a security code so that students or other unauthorized individuals cannot change variables in the table. "SHIFT-LAB" allows the instructor to tell the students certain information such as the purpose of the material, how the material in the lesson relates to other activities in the course, or other work that the student should do in conjunction with the on-line material. "SHIFT-NEXT" brings the instructor to a screen display that allows him/her to decide which section or group of students will actually use the lesson. The instructor is then taken to a screen display which allows him/her to create the units for the lesson and to edit them. Upon choosing to edit the units, the instructor receives an index which outlines the variables to be set for each unit in the lesson. For example, in the "Diatonic Chords" lesson (lesson 9 of Table 1), the instructor sees the index in Table 3.

Table 3

Table Driver Index for "Diatonic Chords" Lesson

1. Competency
2. Sharps or flats in key signature
3. Structure
4. Identification/Construction
5. Test/Drill
6. Triad functions
7. Seventh chord functions
8. Number of voices
9. Clef
10. Inversion

Items 1 and 5 are found in table drivers for all the music theory lessons. With regard to the first item, the instructor can set the number of questions in a unit, the number that the student must answer correctly to achieve mastery, and the maximum average response time allowed to achieve mastery. For item 5, the instructor may choose whether a particular unit will act as a test or as a drill. If the unit acts as a test, the student will only have one chance to respond to a question and will receive minimal feedback for a response. If the unit acts as a drill, the student will have an opportunity to respond as many times as he/she wishes to a question and will get detailed feedback for each response including the correct answer if requested. Mastery of all test units in the lesson is necessary before a student will have mastered the lesson.

Items 2, 4, and 9 of Table 3 are variables common to many of the lesson table drivers. Item 2 has to do with the "weighting" or the probability of a particular key signature appearing in any unit. Item 4 in the "Diatonic Chords" lesson has to do with whether the student is required to identify the Roman Numeral of a chord or whether the student will be required to construct a chord given a Roman Numeral. The terms "construct" and "identify" have similar meanings for other music theory lessons. Item 9 has to do with the "weighting" of the grand staff and the treble, bass, alto, and tenor clefs in each unit.

In item 3, the instructor determines the percentage of chords which will contain an "open" structure and the percentage which will contain a "close" structure. In items 6 and 7, the instructor determines the "weighting" of the various chord function possibilities in a variety of modes. In item 8, the instructor determines whether the program will generate 3-voice or 4-voice chords for a particular unit. In item 10, the instructor determines the "weighting" of root position, first inversion, second inversion, and third inversion chords for each unit.

In all lessons except for the "Transposition" lesson (number 10 of Table 1), the questions are all randomly generated according to an algorithm created by the programmer. In the "Transposition" lesson, however, the melodic patterns used for transposition are all composed by the instructor and stored in computer memory to be randomly selected in each of the units. The reason that no attempt at random generation of melodic patterns is made in this lesson is that in order to allow for the multitude of possibilities for melodic pattern length and intervallic size that different instructors might want would be extremely difficult and not very practical.

Lessons 1 - 5 of Table 1 are currently being rewritten to include table drivers. Hopefully, the new design will allow the music theory lessons to be useful to instructors within a great variety of music theory curricula around the country.

## References

Arenson, M.A. A model for systematic revision of computer-based instruction materials in music theory. Journal of Computer-Based Instruction, 1981, 7, 78-83.

Hofstetter, F. T. Instructional design and curricula impact of computer-based music education. Educational Technology, 1978, 18, 50-53.

# A Pilot Study of a Spanish Version of the PLATO Basic Skills Learning System Mathematics Curriculum

Patricia M. Mullinix - Courses by Computers, Inc.

Ralph T. Heimer - The Pennsylvania State University

## ABSTRACT

Courses by Computers, Inc. translated the Mathematics Curriculum of the Control Data Basic Skills Learning System into Spanish, and a pilot test of the translation was conducted at the Coral Gables High School, Dade County, Florida. The pilot groups consisted of a total of 67 secondary students and adults who had access to eight PLATO terminals, and the participants spent an average of just slightly over 13 hours on-line. All of the students involved in the study considered Spanish to be their native language.

There were three major aspects to the evaluation procedure: 1) A Spanish translation of the ABLE, Level II, was used as a pretest/posttest measure of achievement. The 54 students taking forms A and B of the test showed an average gain in achievement of 1.1 grade levels. Excluding the data of students who entered the program with achievement scores already at the eighth-grade level, 33 students showed an average gain of 1.4 grade levels; 2) Thirteen additional students were pretested and posttested with a Spanish translation of Control Data's *Basic Skills Learning System Mathematics Test*. Based upon this 82-item test an average gain score of slightly better than 11 problems was achieved; 3) Students were queried as to whether the program components were translated clearly, and the responses made by the students were more than 90% positive.

## Introduction

During the year 1980, Courses by Computers, Inc., translated the Mathematics Curriculum of Control Data's Basic Skills Learning System into Spanish. All on-line materials as well as the four supplemental workbooks were translated; three people participated in the process--one of Puerto Rican background, one of Mexican-American background, and one of Cuban background--and all of them were familiar with PLATO'S Basic Skills program.

A pilot study of the Spanish translation was then set up at The Coral Gables High School, Dade County, Florida. There were five classes of basic mathematics scheduled to work at eight PLATO terminals during the day and one adult education class that met two evenings per week. In order to allow each student at least one-half period per day at a terminal, no class was scheduled to contain more than sixteen students. The pilot groups consisted of 67 learners who were able to work at the PLATO terminals from the end of February to the end of May, 1981. The supervision of the project, the training of the staff, and the actual on-line running of the program were all handled through the Computing Center of the Florida State University.

## Demographic Data of Pilot Group

All of the adults and secondary students in the program considered Spanish to be their native language, and most felt their skills in the use of the English language were very weak. Many of the students had entered the United States from Cuba during the year immediately preceding this pilot program.

A demographic questionnaire was entered onto the PLATO system and students responded on-line to these questions at the beginning of the program. Of the 67 students included in the study, demographic data were available for 66. These data are summarized in Table 1.

TABLE 1
Demographic Data Summary

| | # of Students | % of Students |
|---|---|---|
| 1) Native language: | | |
| Spanish | 66 | 100% |
| 2) Native country: | | |
| Cuba | 60 | 90.9% |
| Columbia | 2 | 3.0% |
| Bolivia | 1 | 1.5% |
| Nicaragua | 1 | 1.5% |
| US | 2 | 3.0% |
| 3) Number of years in US: | | |
| 1 or less | 43 | 65.2% |
| 2 | 12 | 18.2% |
| 3 | 2 | 3.0% |
| 4-9 | 1 | 1.5% |
| 10 and over | 8 | 12.1% |
| 4) Number of years of education: | | |
| 8 or less | 6 | 9.1% |
| 9 | 13 | 19.7% |
| 10 | 8 | 12.1% |
| 11 | 24 | 36.4% |
| 12 and over | 15 | 22.7% |
| 5) Degree to which student likes math: | | |
| not at all | 2 | 3.0% |
| a little | 25 | 37.9% |
| somewhat | 17 | 25.8% |
| very much | 22 | 33.3% |
| 6) Degree to which student reads English: | | |
| well | 17 | 25.8% |
| poorly | 49 | 74.2% |
| 7) Degree to which student speaks English: | | |
| well | 12 | 18.2% |
| poorly | 54 | 81.8% |
| 8) Age of student: | | |
| 15 or younger | 3 | 4.5% |
| 16 | 26 | 39.4% |
| 17 | 23 | 34.8% |
| 18 | 7 | 10.6% |
| 19-25 | 1 | 1.5% |
| 26 and over | 6 | 9.1% |

## Data Analysis

At the beginning of the pilot study students were administered the Level I Reading Comprehension Section of the Comprehensive Tests of Basic Skills (CTBS), Espanol, published by McGraw-Hill. Results of the foregoing test indicated only one student score fell below the required level of at least a fourth-grade reading proficiency. There were three additional students who tested as having reading proficiencies contained within the fourth-grade level (scores of 4.6, 4.8, and 4.8). All four of these students did extremely well in the math program in spite of the apparent weakness of their reading skills.

With the permission of the publishers, Harcourt Brace, the Adult Basic Learning Examination (ABLE), Level II, was translated to Spanish and used as the primary vehicle for pretesting and posttesting study achievement. Students in all but one of the classes were given the arithmetic portions of Form B of ABLE II as a pretest and at the end of the program were administered Form A as a posttest. This group included 43 secondary school students plus 11 students from the adult evening class. Analysis of the data from these 54 students showed an average gain in achievement of 1.1 grade levels, from 7.4 to 8.5. A dependent-t test indicated this gain to be statistically significant at the .005 level. (See Table 2.)

TABLE 2
ABLE Test Results For All 54 Students

| | Pretest | Posttest |
|---|---|---|
| Mean | | |
| (raw score) | 30.15 | 36.02 |
| (grade equivalent) | 7.4 | 8.5 |
| Median | | |
| (raw score) | 28 | 37 |
| (grade equivalent) | 7.0 | 8.8 |
| Range | 16 to 48 | 18 to 49 |
| Standard Deviation | 8.49 | 7.57 |

Dependent t Test:

Mean Gain (raw score) 5.87
(grade equivalent) 1.1

$t(53) = 6.577$

t could be this large or larger by chance with $p < .00005$

Note: Raw scores are stated as number of problems correct on the 50-item test. Grade equivalents are taken from the scoring table provided with the test.

Since the Basic Skills program was designed to bring students up to an eighth-grade level of proficiency, gains by students entering the program with skills already at that level would be expected to be very small. For this reason a second analysis was performed excluding the data of students who achieved a grade-level score of 8.0 or greater on the pretest. There were 21 such students, leaving 33 for data analysis. This group showed an average gain of 1.4 grade levels, from 6.5 to 7.9. Again a dependent-t test indicated the gain to be statistically significant at the .005 level. (See Table 3.)

TABLE 3
ABLE Test Results for 33 Students
Below 8th Grade Entry Level

| | Pretest | Posttest |
|---|---|---|
| Mean | | |
| (raw score) | 24.39 | 32.76 |
| (grade equivalent) | 6.5 | 7.6 |
| Median | | |
| (raw score) | 24 | 34 |
| (grade equivalent) | 6.5 | 8.1 |
| Range | 16 to 31 | 18 to 47 |
| Standard Deviation | 4.21 | 7.17 |

Dependent t Test:

| | |
|---|---|
| Mean Gain (raw score) | 8.36 |
| (grade equivalent) | 1.4 |

t(32) = 7.203

t could be this large or larger by chance with p < .00005

Note: Raw scores are stated as number of problems correct on the 50-item test. Grade equivalents are taken from the scoring table provided with the test.

When one realizes that the foregoing students were functioning well below grade level and often showing less than a full grade level increase during a normal school year, the growth experience in this three-month project was exceptionally good. Only four of the 33 students (approximately 12%) failed to show a positive gain score between the pretest and posttest. Fourteen students (better than 42% of the group) showed gains of two full grade levels or better.

The one remaining class that was not administered the ABLE contained 13 students. For this group, the off-line version of Control Data's Basic Skills Learning System Mathematics Test was translated to Spanish by Courses By Computers. Form A was given as a pretest at the beginning of the pilot and Form B was administered at the end of May. This test has not yet been standardized for grade-equivalent scores, but raw scores demonstrated substantial growth in achievement. Based upon this 82-item test, an average gain score of slightly better than 11 problems was achieved, from a mean of 49 on the pretest to 60 on the posttest. A dependent-t test indicated this gain also to be significant at a level of p = .005. (See Table 4.)

TABLE 4
CDC Test Results for 13 Students

| | Pretest | Posttest |
|---|---|---|
| Mean (raw score) | 49.00 | 60.08 |
| Median (raw score) | 53 | 61 |
| Range | 27 to 67 | 35 to 73 |
| Standard Deviation | 12.50 | 8.82 |

Dependent t Test:

Mean Gain (raw score) 11.08

t(12) = 3.726

t could be this large or larger by chance with p = .0029

Note: Raw scores are stated as number of problems correct on the 82-item test.

In addition to the data generated by the administration of off-line pretests and posttests, there were also data collected on-line by the PLATO system. This information, as summarized in Table 5, showed that students spent an average of only 13 hours over the three months actually working in the PLATO lessons. This time does not include time in front of the terminal during which the students were engaged in management or record keeping activities and does not include time spent off-line in workbook exercises. These on-line times ranged between 5½ and 29 hours. It should be noted that both extremes of this range were completion times for the program. Indeed, most of the times at the lower end of the range belonged to students with higher entry levels who were able to complete the program quickly.

TABLE 5
Time Students Spent On-Line

Time is calculated in hours.

All 67 students are included in the sample.

| | |
|---|---|
| Mean | 13.13 |
| Median | 13.17 |
| Range | 5.57 to 29.05 |
| Standard Deviation | 3.81 |
| 95% Confidence Interval | 12.20 to 14.05 |

Of the 73 clusters in the curriculum, students on the average demonstrated mastery of slightly over 60 clusters. This implies that the average level of progress carried students through the topics of whole numbers and fractions and near the end of the decimal strand, leaving them with insufficient time to study ratio/percent and geometry. However, 25 of the entire group of 67 students, or 37%, were able to complete the entire curriculum. (See Table 6.)

TABLE 6
Number of Lessons on Which Students Demonstrated Mastery

There are 73 lessons in the program

All 67 students are included in the sample.

| | |
|---|---|
| Mean | 60.22 (Decimals, cluster 4-5) |
| Median | 65.00 (Ratio/Percent, cluster 4) |
| Range | 34 to 73 (Division, cluster 4 to Completion) |
| Standard Deviation | 12.12 |
| 95% Confidence Interval | 57.27 to 63.18 |

## Conclusions

In order to compare the growth in achievement of the Spanish-speaking students who were involved in the present investigation to that of students who have previously studied the English version of the Basic Skills Mathematics Curriculum, it is necessary to summarize the results of several studies of the English version. An evaluation conducted by Florida State University in three Florida high schools during the 1978-1979 school year, using ABLE-Level II, resulted in the following summary data:

- The overall median gain for all students was 1.5 grade equivalents after spending an average of 20 hours in the PLATO math lessons.
- On the average, students spend 13 hours in the PLATO math lessons to achieve 1.0 grade equivalent gain.

At the Baltimore Adult Learning Center, in 1978, students were also pretested and posttested using the ABLE with the following results:

- The first 43 students completing the program showed a mean gain of just over 2 grade levels after an average of 24 hours in the PLATO math curriculum.

It is evident, therefore, that the results of using the Spanish translation are very similar to those involving the English version. Whether examining the progress of secondary students or adults, English-speaking learners or Spanish-speaking, it appears that an average of approximately 13 hours on-line in the math curriculum can produce growth of one full grade equivalent.

---

Mullinix, P.; Grynovicki, J.; and Bowen, C. The Effects of the PLATO Basic Skills Learning System in the Baltimore Adult Learning Center--A Preliminary Report. Baltimore, March 1979.

Qualls, J. and Brown, B. FSU PLATO Project: Basic Skills Math for Florida High Schools. Tallahassee, Florida State University, 1979.

# NETWORKED MICROCOMPUTERS

## PROBLEMS AND PROSPECTS IN UNIVERSITY USE

Paul Lorton, Jr. - University of San Francisco

John Eaton - London Business School

## ABSTRACT

Networked microcomputers have become one possible choice for student oriented computer systems. Using the initial experience of the London Business School with a Nestar Network of Apple II's as a base, this discussion addresses the problems associated with installing such networks in a school setting. Since networked microcomputers often replace an existing timesharing system, the differences between timesharing and networked microcomputers is of particular interest. Possible solutions to major problem areas of hardware, software and staff are indicated and the ultimate advantages of networked microcomputers are mentioned.

The London Business School (LBS) acquired a Zynar/Nestar File Server Network and several APPLE Europlus microcomputers to replace a Hewlett Packard 2000 Series Timeshared BASIC System. The Network has been in use as a general student computing resource since Fall Term, 1981.

All new computing resources present problems before their promise is realized. The purpose of this discussion is to outline some of these problems and some of their solutions. Some of the problems are general to all new computing resources in education, some are related to the use of microcomputers. The most important and interesting are those which have to do with networks of microcomputers.

## NEW COMPUTING RESOURCE

The main problem with a new computing resource is almost always software. In some cases this can be as mild and pleasant as simply choosing what to promote among the users from a wide range of available programs. In the case of the LBS Network, the software problem was (and will remain for some time) how to replicate the features which most staff and some students had grown to expect.

A network of microcomputers is not, nor was it intended to be, a timesharing system. However, the staff and experienced students expected the new system to continue the developmental path of the old. Some features and some specific programs had to be transfered from the timesharing system. Software as straightforward as a system for locating possible programs (a Menu and/or Library system) or as complex as the Interactive Data Analysis package which had been developed on the Hewlett Packard system over a period of more than seven years needed to be implemented on the network.

A further problem area, associated with making available programs useful on a new computing resource, was defining exactly what is the easiest, most comfortable procedure to use for users with a wide range of sophistication. At this point, most user interfaces for timesharing systems are well developed out of long experience. Shifting from one timesharing system to another always presents some problems. But the problems are usually of a very minor and superficial nature - what new command does one issue to access an old feature. With a new computing resource that is also a new approach to computing, the task is not quite as simple. One cannot simply learn a new command sequence for the parallel feature simply does not exist. An example of this are questions about how one accesses the print utility on the new system. Going from one sophisticated timesharing system to another, this is an answerable question. Going to a network of microcomputers, the answer is a much more difficult one, "There is no print utility!"

The hardware of a new computing resource always represents a potential problem for service and maintenance. With some experience with any system, a sense of how difficult ordinary problems are to fix develops. Anytime the hardware configuration is changed, one has to learn a new set of parameters for investing resources in service and maintenance. With new hardware which is also a new approach, all previous experience is of little use. Even a relatively simple concept such as "down time" will have a new meaning.

### MICROCOMPUTERS

The fact that the new computing resource was microcomputers added a further level of difficulties because microcomputers are a different computing environment in many ways. Some areas in which this sort of problem arose were software, perception and the display.

The software problem related to microcomputers per se (rather than just part of the new computing resource problem), has to do with what is available for microcomputers. There is a lot of software available for microcomputers for there are hundreds of "cottages" producing dribs and drabs. How can one locate it? How can one evaluate it?

With larger systems there are supporting users groups (often instigated by the manufacturer) to help in this effort. For the microcomputer industry, the user groups are more often locally stimulated and parochially oriented. The "personal computer" journals are a good source but, for reasons of broadest appeal in the market place, their evaluation efforts tend to be on the larger and more generally used software: games, simple business packages and word processing. The software market, notwithstanding the tremendous activity, is not mature enough to provide systematic information on specialized applications.

The perception of microcomputers is another problem unique to a microcomputing environment. For those with timesharing experience, the microcomputer network station looks like a timesharing terminal. It has a display and a keyboard so how is it, how can it be, different. It is vastly different. A timesharing system works very hard to keep itself away from the user - the microcomputer's operating system does very little to protect itself. On a timesharing system, much user assistance is available - on a microcomputer, very little user assistance is built in.

Another problem, related to microcomputers per se, is the display. Microcomputers, especially the APPLE, are very much oriented toward the video screen and as an output device and very much oriented toward graphics on that screen. The APPLE is especially good at graphics and especially poor at text - 24 lines of 40 upper case only characters. Programs developed for an 80 character wide timesharing system terminal and which were perfectly useable if the text scrolled on a printing terminal, require substantial reorientation to make them appropriate for the microcomputer's display.

### NETWORKS OF MICROCOMPUTERS

The fact that the installed system was not only microcomputer based but a network of microcomputers added another level of problems to those already discussed. Networks present many advantages but the initial disadvantage is that the main operating orientation of all but a very few computers is to stand alone. This is true of large IBM or ICL systems and true of APPLES and ACORN ATOMS. While networks are increasingly available, they are a new focus for computing and a focus in search of a direction.

The possible directions a network of computers can have include one or all of: communications, shared resources, expanded on-line storage, central control distributed processing and/or distributed input/output of information. For an educational network, shared resources, expanded on-line storage and some central control seem the most appropriate for an initial direction.

The problem with all three, shared resources, expanded on-line storage and central control, is that timesharing systems offer the same set. However, the timesharing system is not an entirely appropriate model for networked microcomputers.

### DEVELOPING SOLUTIONS

The problems which surfaced in the areas discussed above grouped themselves into several foci for solution: hardware, software, staff interface, student interface and global difficulties.

1) Hardware:

Although hardware providers are often quite eager to maintain new equipment, especially for installations very early in the product's life, there was some concern about the ability of the suppliers to maintain the equipment. Thus far, the hardware has been well maintained and serviced. Moreover, the system is fairly modular so that, even with the central disk resource out of service, some computing is possible. The loss of individual microcomputers from the system has, as would be imagined, no more impact than the loss of an individual terminal from a timesharing system.

However, because the terminals are microcomputers, basically designed to be stand-alone computing systems, what kind of hardware to include with each station is a problem. The hardware provider would like to see as many optional extras as possible - language cards, special display cards (color PAL cards, 80 column - lower case cards, disk controllers, printer controllers - all with associated peripherals). The benefit of a network for education should, however, be minimizing the array of special features installed in each microcomputer.

Thus far the solution has been to stay with the minimum hardware configuration at each station and try to make software adjustments to overcome the hardware limitations at individual stations. This response means "making do" with upper case only word processing as well as converting useful programs which depend on other languages (e.g., APPLE's Integer Basic) to run on stations without these languages (e.g. compiling the integer basic programs).

2) Software:

Part of the software problem is solved by the orientation to hardware enhancements described above. This, at least, placed some limit on which programs to make available, and which to choose.

3) Staff Interface:

Making the staff aware of the changed environment is a problem for they are more prone to feel it is exactly the same as the old environment than the students. We all tend to look, because of our experience over the last 15 years, on new systems as part of a gradual evolution. With microcomputer networks, the shift is abrupt and into an area with many unknowns. Answering questions which start with, "We used to be able to do that ..." is quite difficult.

Moreover, the students will not be told the advantages of the new system if the staff does not know them. The solution has been to provide brief discussions of various features and to encourage creative use of the new tools. The most promising and unanticipated trend has been for staff to acquire their own stand-alone microcomputers and use these to develop an awareness of the potential as well as applications for the students.

4) Student Interface:

At present the solution to the student interface problem has been to provide as flexible a menu program as possible and to all but insist that the students report any needs this menu system does not meet. It is a long, slow process which as the student population alternates between those who grow in sophistication and are soon replaced by those who start being totally naive.

5) Global Difficulties:

The two most global difficulties are reorienting programs to the display and addressing the differences in network versus timesharing disk access. For the former, providing good models which represent conformity to some standards seems most fruitful. For the latter problem, the answer is known but the solution is difficult. On a timesharing system the time to access the disk is quite little as it is accessed by the central processor. With a network, transmission of the request is over the network and can be severely slowed by competing requests. Further, initially loading a program into a student station can be quite slow for the whole program must be transmitted across the network before the micro starts running. Redesigning programs so that they phase their requests to the file server will be a long, slow process toward solving this difficulty. However, in the solution to this problem will be some of the answers to questions on the real differences and advantages of networks over timesharing systems.

ADVANTAGES/BENEFITS

A discussion which focusses on problems is going to be a little gloomy at best. However, networked microcomputers represent a substantial innovation as an educational tool. Certainly the innovation is as great as the first timeshared BASIC systems represented. Then, taking proper advantage of the tool in education was a learning process for the staff and the students. The same is true of networked microcomputers - taking proper advantage of their strengths will enhance the educational experience of students as they learn about both using computers as tools and as computers.

One of the main advantages of the networked microcomputer is that the processing is distributed. There is no need for any real "down time" for the whole system, for example. If the central disk is down for back-up or maintenance, the individual station can still function and can be re-connected to the system when it is operating again. Further, work can be done on any stand-alone micro and easily merged into the network. Several faculty are already taking advantage of this to develop applications on their personal Apples which they transfer onto the central file server by bringing floppy diskettes from home.

## FUTURE CONCERNS AND PROSPECTS

The problem with the problems is that they have been seen before - yet, because of the new environment, their solution will yield as large a difference in how the system looks and operates as timesharing differed from batch processing. The prospects are quite good. In addition to the features and directions possible with networks already mentioned (communications, shared resources, etc.) the networks can also include linking personally owned personal computers into the system. This will become a real possibility in education quite soon. As such projects as the BBC's microcomputer and television series encourage more homes to have personal computers, educational environments will be all but forced to integrate the use of these machines into whatever network they use.

The concern is that networks will be forced to emulate a timesharing environment because it is easily done and the model is well understood. In the long run, this sort of solution will not work, but the effort of trying to make it work may retard the development of networked microcomputers into computing systems with their unique strengths emphasized and their inherent weaknesses reduced.

# TOWARDS PORTABLE MICROCOMPUTER COURSEWARE

Norman F. Salt
The Ontario Institute for Studies in Education
252 Bloor Street West
Toronto, Ontario, Canada M5S 1V6

## ABSTRACT

There is currently a need to make good courseware available on microcomputers. In the past, various strategies have been used to implement courseware delivery systems. Each strategy has its own strengths and weaknesses. In this paper, an approach is recommended in which a significant strength is portability. A case study involving the development of a portable interpreter for the CAN coursewriting language is described. The interpreter is written in Pascal. The implementation includes portable graphics functions.

## THE SHORTAGE OF COURSEWARE

In the field of computer assisted instruction, there seems to be a shortage of good courseware. This shortage is exacerbated because most of the best courseware is not portable. Typically the courseware developed for large time-sharing systems is not easily transferable to microcomputers. When it is transferable, it is frequently limited to one type of microcomputer.

The flourish of activity in the production of instructional programs specifically for microcomputers has done little to increase significantly the pool of useful courseware, particularly in areas other than the basic skills. Many of the myriad of recent programs are neither good nor portable. Some programs have been written with no consideration for pedagogical principles and instructional strategies. This may be due in part to the fact that authors are hampered by the lack of an adequate coursewriting language. Often they are forced to use a general-purpose computer language or, at best, a coursewriting language with limited facilities. When good courseware does emerge for a particular machine, it is rarely executable on another brand of microcomputer.

Even the abundant amount of microcomputer courseware written in BASIC is not particularly portable. A case study by Cavin and Cavin (1979) demonstrates some of the problems of transferring BASIC courseware. Although the BASIC language is widely available on microcomputers, each microcomputer seems to support a different version. A comparison of the versions ("Basics Comparisons Chart", 1980) indicates a tremendous variety of dialects, especially in the areas of graphics and language extensions. The hope for standardization is remote.

## COURSEWRITING LANGUAGES ON MICROCOMPUTERS

There are numerous coursewriting languages in use. These include PILOT, TUTOR, Coursewriter, PLANIT, and CAN. It is unlikely that any single language is best for all instructional applications. It would be ideal if a few of the most common coursewriting languages were generally available on microcomputers. Then authors could choose the most suitable language for a particular application, and teachers could select lessons from a wider range of courseware. Although this ideal has not yet been realized, there are some developments in this direction.

Van der Mast (1978, 1979) implemented a courseware delivery system using a proposed universal intermediate assembly-type language (CAIASCO). Theoretically, any coursewriting language can be translated into the intermediate language. Such translation is facilitated by the fact that the intermediate code is open-ended; new operations can be added to accommodate the idiosyncracies of any present or future coursewriting languages. The intermediate code is processed by an interpreter (CAIMIC) on a microcomputer. In van der Mast's implementation, the interpreter is written in UCSD Pascal (Bowles, 1980). This version of Pascal is available on many microcomputers. Van der Mast reports translation of PLANIT into the intermediate code. Work on a translator for a subset of the TUTOR language is in progress. The advantage of using an intermediate language is that it can be interpreted efficiently and quickly. Furthermore, with a portable interpreter, the intermediate code can be interpreted on a variety of different microcomputers. A disadvantage of the strategy is the effort required to translate new coursewriting constructs into the intermediate code. Subtle differences in actions required by similar high level instructions can be troublesome to translate into a universal code. Either the translation is awkward (or impossible) or the number of intermediate language operation codes becomes unreasonably large.

Mundie (1980) provides an approach that obviates the need for a new interpreter. His approach is to translate a dialect of the PILOT language directly into Pascal. Once the courseware is translated into Pascal, it can be interpreted or compiled by existing software. This approach is reasonable for a relatively simple language whose constructs are easily mapped into an available high level language. In the case of the PILOT dialect to Pascal, it was reported that the translation was effected quickly. However, with a more complex coursewriting language, especially one that is structurally or functionally dissimilar to the target language, the translation effort could be extremely frustrating or even impossible.

Another approach to providing courseware on microcomputers is offered by Garton and Silver (1981). They describe the translation of TUTOR courseware into microTUTOR courseware. To the extent that this process can be automated, the approach - at least for TUTOR - is promising. Unfortunately, the process may be difficult to automate for some parts of the more successful and complex types of lessons such as the GUIDO lessons described by Garton and Silver.

Merrill, Hawkins, and Hendricks (1981) report the implementation of many features of the TICCIT system on a microcomputer. Complex constructs of TICCIT are made available on an Apple II microcomputer. However, while the study demonstrates what is feasible, the routines are not portable. They are written in a dialect of BASIC that is used on the Apple II (APPLESOFT).

## AN APPROACH FOR NEW TECHNOLOGY

Most efforts to make courseware available on microcomputers involve some form of translation from the coursewriting language to some other computer language. In the context of the early microcomputers, an important advantage of this strategy has been the small size of the interpreter required to process the translated language. However, the cogency of this advantage is diminishing rapidly with the advent of microcomputers using 16-bit architecture. Current and future hardware products with relatively inexpensive main memory make it feasible to interpret complex coursewriting languages directly rather than to translate them into intermediate codes. In the Canadian province of Ontario, the proposed Canadian Educational Microcomputer (Buckler, 1981) is likely to support relatively large interpreters. With this computer and similar ones that will be in the classrooms of tomorrow, the disadvantages of using large interpreters become less important. With the new generations of microcomputers, there should be sufficent main memory for the interpreter. Moreover, with a fast, dedicated microcomputer, unacceptable delays in lesson delivery are unlikely.

A cursory examination of the effort required to make L coursewriting languages available on M machines suggests that LxM interpreters would be required. Obviously the effort required for the development of so many interpreters is not about to occur. However, a careful examination of the state of currently available microcomputer software leads to a more reasonable option. If the interpreter were written in a language available on all popular classroom microcomputers, then only one interpreter would be required for each coursewriting language.

## THE PASCAL SOLUTION

If portable interpreters are to be written for microcomputers, there are only two microcomputer languages at present that warrant consideration for purposes of implementation: BASIC and Pascal.

The most common microcomputer language today is BASIC. It was the only high level language supported by the first microcomputers introduced in 1975. In a recent survey of 59 micro systems, BASIC was available on 54 (Fox, 1981). BASIC is short for for Beginners All-purpose Symbolic Instruction Code. As the acronym suggests, new programming students seem to learn the language quickly. However, some of the features of BASIC that facilitate rapid programming of short programs (e.g., GOTO and GOSUB) lead to development of programs with no coherent structure. The writing of complex interpreters in BASIC is not recommended unless a structured version of BASIC is used. Unfortunately, the very use of a structured version reduces the portability of the final product. In fact, writing in almost any version of BASIC leads to serious concerns about portability.

The current alternative to BASIC for writing a portable interpreter is the Pascal language. The language is already available on 33 of the 59 micro systems surveyed by Fox. Furthermore, there seems to be a commitment to Pascal on the part of all the major microcomputer manufacturers. The language was designed with the intention of correcting the faults of languages that existed in the 1960's. It provides constructs for developing well-structured, modular programs. Moreover, in contrast to the lack of standardization in BASIC, there is a definitive description for what has become known as standard Pascal (Jensen & Wirth, 1974). Although there are various versions of Pascal, including a newly developing standard, they generally involve extensions to standard Pascal. The standard Pascal remains embedded in the new versions. Some extensions (e.g., a string type) do make programming easier, but with some extra effort a programmer can code without using the extensions.

In writing with portability in mind, the programmer must be aware of certain features of the Pascal language. There are machine-dependent features of standard Pascal (Grogono, 1980, pp. 355-356) that should be avoided by the programmer. However, one non-standard feature of Pascal should be used in the development of an efficient interpreter. The feature is random access file handling. The use of this non-standard feature means that slightly different versions of the interpreter may have to be generated for different types of machines. Fortunately, the differences can be isolated in a single Pascal procedure and limited to a few lines of source code. There are also other variations among implementations of standard Pascal, but they are minor.

## A CASE STUDY

With the intention of investigating problems involved with developing and implementing a portable interpreter for a coursewriting language, the author designed and coded such an interpreter. The coursewriting language selected for the implementation was CAN-8$^{tm}$ (Olivier, 1981). CAN-8 is a recent version of the upwardly compatible CAN family of coursewriting languages. CAN was developed at the Ontario Institute for Studies in Education in 1967 and now enjoys wide use throughout Ontario (McLean & Olivier, 1980). The results of a survey by Wang (1978) indicate the availability of about 1500 hours of instructional material in the CAN language in the mid-1970's. This compares with approximately 5000 hours of TUTOR material and 3000 hours of Coursewriter material.

Except for a procedure to randomly access the CAN source file, the interpreter was written in standard Pascal. The lines of code that may require modification depending on the Pascal compiler are few. An example of such a modification is the built-in Pascal function "arctan"; it must be changed to "atan" to execute on the Apple microcomputer. The procedures in the interpreter are ordered for efficient segmentation in a UCSD Pascal environment. Program notes indicate where segment statements should be located if segmentation is necessary.

A few of the CAN-8 features are not handled by the Pascal interpreter. One omission is a subset of the display language called "text modifiers". These modifiers have the effect of altering the terminal display in various ways. Many of the omitted text modifiers involve the display capabilities of a particular terminal. Since these capabilities vary considerably, different coding would be required for each terminal type. Such coding is certainly possible, but it would not be portable. The most common text modifiers are not terminal dependent and some have been implemented. These include displaying a student's name, displaying the contents of variables and buffers, suppressing carriage returns and line feeds, and inserting multiple line feeds.

Another CAN feature that is not handled is input and output to peripheral devices other than the student terminal. Hence, this implementation cannot operate devices such as audio-tape recorders,

---

CAN-8 is a registered trademark of Homecom Learning Systems, Limited.

slide projectors, videotape players, and videodisc players. It cannot create disk files with student records. The control of some peripheral devices is hampered by the wide variation in control signals required to operate different devices. The facility to create disk files was omitted because no standard format exists for disk file records for a stand-alone system. The reading and writing of disk files could be incorporated easily when standards are defined. In their work with TICCIT, Merrill et al. (1981) reported no problems in implementing a record keeping function on a microcomputer.

A further omission is the option to terminate a student's response based on time and length of response criteria. Implementation of this facility is machine dependent. It could be implemented on many machines, but it cannot be coded in standard Pascal.

In spite of the omissions, the Pascal interpreter provides most of the features required for computer assisted instruction on a microcomputer. All of the powerful CAN features such as the complex answer matching algorithms and multiple branching are included. The few missing features could be added without difficulty if and when desired.

## CAN GRAPHICS

The CAN-8 language contains a set of commands for graphing. The commands are based on the Picture Description Instructions (PDI) for the Telidon Videotex system (Bown, O'Brien, Sawchuk, & Storey, 1979). Among facilities provided by the graphics commands are those for drawing lines, arcs, and polygons. Lines may be solid or a series of dots and dashes. Functions are available for filling enclosed figures with a solid colour or a grid pattern. Transformations of scaling, rotation, and translation are included. For colour terminals, eight colours are specified.

Efficient implementation of the graphics commands on non-Telidon terminals is problematic. There are no industry graphics standards to which microcomputer manufacturers adhere. Yet the graphics commands are an important feature of CAN-8. To omit them from an implementation is to reduce the usefulness of all courseware containing graphics. The method of providing the most efficient CAN graphics on microcomputers is to write a different graphics interpreter for each type of display hardware. Each interpreter could exploit the graphics capabilities of the particular terminal.

The author chose another option for implementing the microcomputer graphics. At the cost of not utilizing the manufacturers' graphics packages, portable Pascal procedures were written to implement CAN-8 graphics. Published algorithms were used to implement the more complex procedures. Three primitive functions were required by the implementation: initialization of the screen for graphics, termination of the graphics mode, and plotting a point specified by its coordinates and an attribute. The attribute is a colour, a grey-scale intensity, or a blinking value. The coding for each primitive varies from microcomputer to microcomputer; it depends on the type of video display device. However, the amount of coding to be changed is very little. For many implementations, one line of coding is required for each of the initialization and termination primitives, and a dozen lines are required for the plotting primitive. The only other variation from machine to machine is a change in the values of constants that contain scaling information to transform points to the target screen. This scaling data ensures that all shapes from the virtual CAN screen appear on the terminal screen without distortion or clipping. The virtual CAN screen is 1024 points wide. Since most current terminal screens contain only a fraction of the points represented on the virtual CAN screen, resolution is usually dependent on the terminal screen.

The Pascal implementation of the CAN-8 graphics does occupy more main storage and may operate more slowly than the manufacturer's graphics package. Nevertheless, it is reasonably portable and does provide the full graphics capabilities of CAN-8. An example of a graphics display produced with CAN-8 commands is shown in Figure 1. The Figure provides a static illustration of the interactive graphics capabilities that can be made available on a variety of microcomputers. If the microcomputer supports Pascal and offers the graphics capability to plot a point at given coordinates, then the microcomputer will support CAN-8 graphics commands.

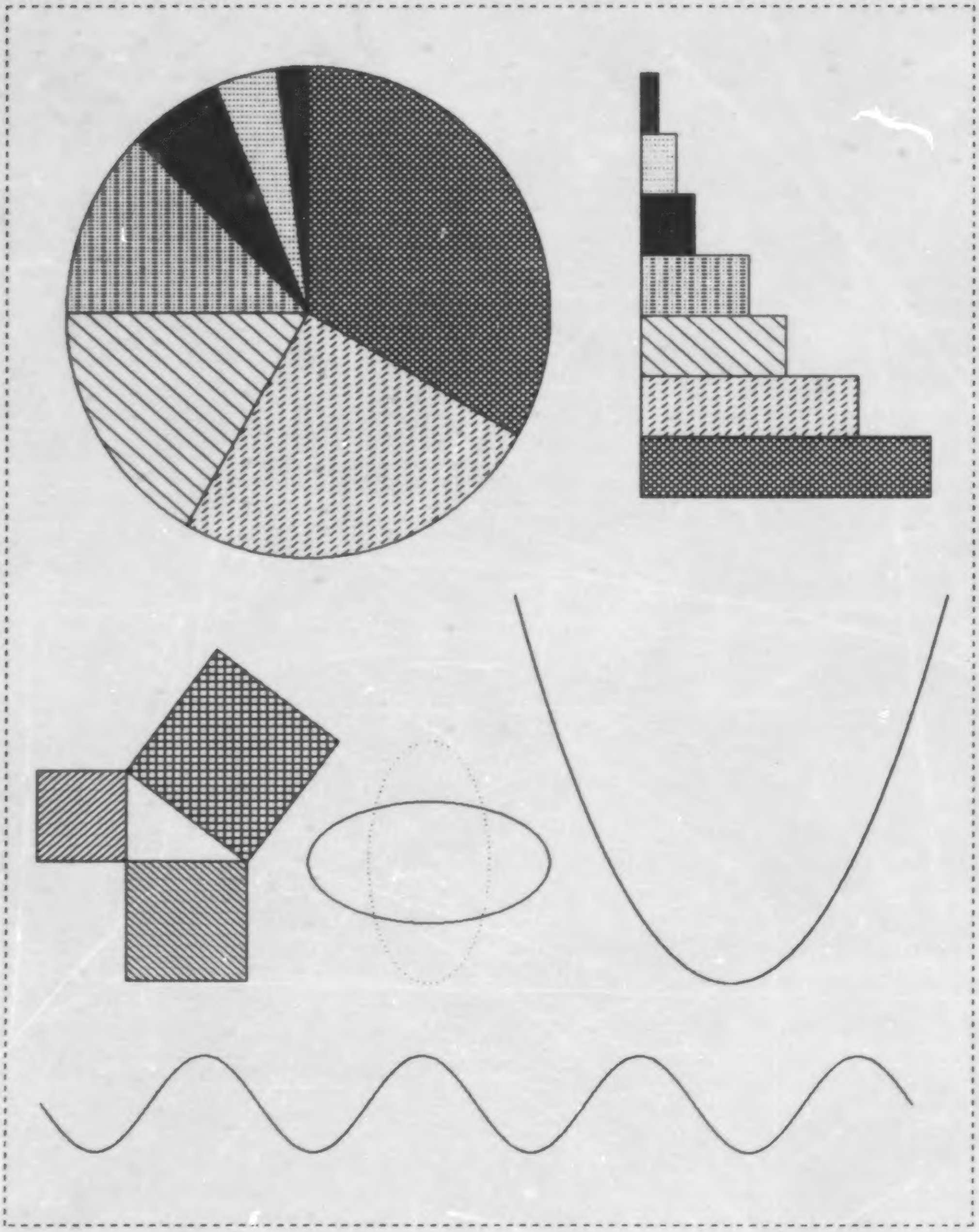


Figure 1. A display generated by CAN-8 graphics commands.

## CONCLUSION

It has been observed that among the problems plaguing CAI during its twenty-year history, the shortage of good courseware has persisted with the advent of microcomputers (Braun, 1979). While this shortage persists, there is reason to offer the meagre quantities of good courseware to the growing numbers of students who have access to microcomputers. A technical strategy for making the courseware accessible is to develop portable interpreters. If proprietary concerns could be resolved, then good courseware developed and used extensively on large machines could be made available on microcomputers. Additionally, new courseware using the facilities of more complex coursewriting languages could be developed for use on microcomputers. In summary, a few carefully-written, portable interpreters for the major coursewriting languages could open the possibility of providing some good courseware for students on any of the popular microcomputers.

## REFERENCES

Basics comparison chart. Creative Computing, July 1980, pp. 28-29.

Bowles, K. L. Beginner's guide for the UCSD Pascal system. Peterborough, New Hampshire: Byte Books, 1980.

Bown, H. G., O'Brien, C. D., Sawchuk, W., & Storey, J. R. Picture description instructions PDI for the Telidon Videotex system (CRC Technical Note No. 699-E). Ottawa: Communications Research Centre, 1979.

Braun, L. Computers and education - the promise and the fulfillment. Apple, 1979, 1(1), 2-5.

Buckler, G. Ontario plans school micros. Computing Canada, September 3, 1981, pp. 1; 11.

Cavin, C. S., & Cavin, E. D. A case study on an inter-computer transfer of CAI programs. AEDS Monitor, 1979, 17(13,14,15), 14-15.

Fox, T. Let the buyer compare. Interface Age, January 1981, pp. 76-78; 81-84.

Garton, R., & Silver, J. Approaches to converting PLATO courseware to MicroTUTOR: A case study. Conference Proceedings of the Association for the Development of Computer-Based Instructional Systems, 1981, 162-166.

Grogono, P. Programming in Pascal (Rev. ed.). Reading, Massachusetts: Addison-Wesley, 1980.

Jensen, K., & Wirth, N. Pascal user manual and report (2nd ed.). New York: Springer-Verlag, 1974.

van der Mast, C. A modular CAI system. In D. Brook & P. Race (Eds.), Aspects of Educational Technology Volume XII: Educational Technology in a Changing World. London: Kogan Page, 1978.

van der Mast, C. The presentation of courseware by microcomputers using a modular CAI system. ACM Sigcue Bulletin, 1979, 13(4), 2-9.

McLean, R. S., & Olivier, W. P. Computer assisted learning in Ontario. In R. Lewis & E. D. Tagg (Eds.), Computer Assisted Learning: Scope, Progress and Limits. London: Heinemann Educational Books, 1980.

Merrill,P.F., Hawkins,A., & Hendricks,H.H. TICCIT for the Apple. Conference Proceedings of the Association for the Development of Computer-Based Instructional Systems, 1981, 162-166.

Mundie, D. PILOT/P: Implementing a high level language in a hurry. Byte, July 1980, pp. 154; 156; 158; 160; 162; 164; 166; 168; 170.

Olivier, W. P. CAN-8 language reference manual. Toronto: The Ontario Institute for Studies in Education, 1981.

Wang, A. C. (Ed.) Index to computer based learning 1978 edition. Milwaukee: Instructional Media Laboratory, The University of Wisconsin-Milwaukee, 1978.

# The CLEF Project:
# Learning French on Colour Micros

**Glyn Holmes - University of Western Ontario**

**Marilyn Kidd - Huron College**

ABSTRACT.
The CLEF Project was begun in 1976 at the University of Western Ontario, Canada for the purpose of creating and testing CAL materials for use in the learning of French. The pedagogical objectives of the project are to create a full complement of grammar lessons and exercises for beginning and intermediate students for use as an adjunct to regular classroom instruction. The technical objectives consist in the creation of software that is easily portable using a machine that offers user convenience and versatility at a purchase price that is reasonable. In designing courseware, care has been taken to produce quality pedagogical materials that take advantage of the unique technical features provided by the colour microprocessor. Lesson strategies and details of courseware design developed specifically for this medium are discussed with examples.

## I. Computer-Assisted Language Instruction: A Brief Overview

In the late 1960's the Universities of Stanford (Suppes, 1968) and New York at Stony Brook (Adams, 1969) introduced programs designed to test the possible application of computers to the learning of foreign languages. The ultimate aim in both cases was to use the computer as the major, if not the sole source, of grammar instruction. A somewhat less ambitious project begun in this initial period at Dartmouth College (Allen, 1971) suggested that the computer might better serve merely as a supplement to traditional classroom instruction.

Once the concept of Computer-Assisted Language Instruction (CALI) had been introduced, subsequent researchers threw their efforts into creating systems that would be more technically sophisticated. The 1970's saw a number of programs created in which technological innovation predominated. To the basic teletype or CRT were added various peripherals: audiocassettes, light sensitive pens, random access slide projectors. Projects such as PLATO at the University of Illinois (Marty, 1975) or FRAND at the University of Alberta (McEwen, 1977) are examples of the attempts to expand the technological versatility of the computer.

Certainly, these experiments were laudable endeavours to increase the scope and effectiveness of the new medium and yet somehow, in the rush to introduce a host of new technological features, pedagogical considerations were, as during the late 60's, sometimes relegated to second place. After the initial euphoria had abated somewhat, it became apparent that there were in fact many drawbacks to such a system. In addition to complications arising from the necessity to co-ordinate several media successfully, there were queries concerning cost-effectiveness and the realization that many applications merely duplicated instruction that could be performed better and cheaper by other means. What the technicians had shown to be possible proved not always very feasible, and the feeling that somehow the computer had not lived up to its expectations prevailed in some quarters. (Kopstein, 1973)

Yet a seed had been sown and, after a period of disillusionment, CALI was to be revitalised, this time by the people closest to it - the language teachers. They had glimpsed the basic potential of the computer and now set about to realize it in a more modest way. Now programs such as DECU and TUCO at Ohio State (Taylor, 1978), or the Russian programs at

New York State University College at Cortland (Olmsted, 1975 ) had pedagogical effectiveness as a first priority and technological impressiveness was given a secondary importance. Large, expensive projects were often replaced by smaller endeavours such as MONIQUE at the University of Richmond (Terry, 1978), or CARLOS at West Chester College (Smith, 1976), designed to fulfill the specific needs of a particular course. Sometimes they were the work of one man, and not infrequently, as in the EGAPO project at Algonquin College, Ottawa, the language teacher also assumed the role of computer-programmer. (Villeneuve, 1980). At last educators were looking not just at what the computer could do, but at what it could do best.

This trend seems to be continuing. Recent attempts in CALI are concentrating on developing the computer for use in those areas of language learning in which it can make a unique contribution. Such is the case with the CLEF project.

## 2. THE CLEF PROJECT

### 2.1 Conception

In 1976 the Language Laboratories at the University of Western Ontario undertook a modest experiment in Computer-Assisted Language Instruction. The initial program, created by a part-time French instructor and a computer programmer, was a vocabulary drill for Introductory French, written in FLEX, a dialect of FORTRAN. The hardware configuration involved four teletype terminals attached to the university's timesharing computer, a PDP-10. Though the initial programs were decidely modest, they met with a certain success, and soon a number of other language instructors were experimenting with various vocabulary and grammar-based programs in French, German, Italian and Remedial English. In 1978 a major project was launched involving a team of French instructors from the University of Western Ontario and the London Board of Education, and whose aim was the creation of a series of grammar-based lessons for French. The work of the team of coursewriters progressed so well that within a year we had joined forces with teams at the Universities of Waterloo and Guelph, and had been awarded a significant grant by the Ontario Ministry of Colleges and Universities. This was the beginning of the inter-university project called CLEF (Computer-assisted Learning Exercises for French).

The main thrust of the research was aimed at the creation of courseware materials. This activity was centred at U.W.O. and Guelph. Two years ago, the University of Calgary became the third member of this triumvirate and is actively participating in the development of lesson materials.

### 2.2. Objectives

The fundamental aim of the CLEF project is the creation of two complete series of French CAL lessons, one at the introductory level, the second at the intermediate level. Each series will contain drill and practice exercises on all major grammatical points encountered by students at that level. Though the precise uses to which the lessons might be put will vary according to the needs and the imagination of individual users, the CLEF lessons are conceived as an out-of-class supplement to the usual instructional process involving teacher and textbook.

In addition to these general aims, there are a number of more specific educational and technical objectives.

#### (1) educational

The pedagogical materials are, of course, the heart of any CAL system though it has rightly been said that, particularly in some early CAI experiments, the actual preparation of the educational materials was relegated to secondary importance. (Meredith, 1971). In the CLEF project, this has definitely not been the case, and educational concerns have been of paramount importance.

Firstly, we wanted our materials to be text-independent. Whilst some CAI programs have used materials taken wholesale (Suppes, 1968; Adams 1969), or adapted from a textbook, this of necessity limits their application and can, when texts are changed, leave one with an expensive, but outmoded set of lessons.

Our second concern was a corollary of the first: we wanted our lessons to be useable at all educational levels, from the younger learners in the primary grades, to university students or adult learners who might not necessarily be following a structured French program. To ensure a homogeneity of approach to this variety of levels, the coursewriting team includes instructors from all the appropriate levels.

One of our principal educational goals has been the creation of quality pedagogical materials that are appropriate to the medium of computer-assisted learning. There exist many excellent programs for the learning of French, in the form of textbooks, audio tapes and filmed materials. Yet it is perhaps not yet universally recognized that materials and approaches that are used for one medium are not likely to be as successful if transferred to another medium.

(ii) technical

The members of the project were determined that the system chosen should facilitate, to the greatest possible degree, the widespread distribution of the materials created. Such objectives militated against the use of a mainframe configuration. Many programs have seen little or no use outside the institution in which they were created simply because of their dependence upon a particular, and expensive mainframe system. Attempted solutions such as portable terminals, or transporting users to the mainframe location (McEwen, 1977) are either costly, inconvenient, or both. A second objection to the mainframe is the fact that when the system is down, all terminals cease to function until an independent body, usually the Computing Centre, can rectify the problem. Such situations can, as we have observed, lead to frustration and even disenchantment with the medium, perhaps more frequently in the case of Humanities students who have generally less exposure to machinery and its imperfections than Science students.

The microcomputer therefore offered substantial advantages over the mainframe: it was cheap, highly transportable and allowed for quick and easy distribution of materials. We opted for 32K Compucolor II, and Intecolor 3621 colour microcomputers. Both machines, which are made by the same parent company and are quite compatible, have internal disk-drive that can access a 51K bank of data files. The only specialized piece of hardware we required was a chip containing small character ASCII and accented French characters, which replaces the standard chip supplied by I.S.C. (Intelligent Systems Corporation).

It was this same desire to encourage widespread distribution and use of materials that prompted us not to utilize specialized software. The choice of programming language, like that of hardware, can have an inhibiting effect on the use of courseware. Problems of this nature have been recognized by such marketing agencies as CONDUIT which requires all programs to be written in FORTRAN or BASIC, and the Canadian National Research Council has been working to produce a standard CAI language, NATAL-74 (Brahan, 1978). We therefore used the language resident in the machine, BASIC, a variation of the same dialect of BASIC that runs on most microprocessors. The problems of adaptation to other micros are thereby minimized.

### 2.3 The CLEF Lessons: Level I

(i) Grammar Presentation

The level I lessons, designed for use by beginning students of French, are introduced by an optional, brief review of the appropriate grammar point. At this stage it is quite unfeasible to use French as the language of instruction. Consequently, we used English to explain grammatical structures. In addition, when it was deemed appropriate, we adopted a contrastive (English-French) approach to such questions as the position of adjectives, the use of prepositions, etc.

It was anticipated that not every student would either want or require reinforcement of grammatical materials to which he had been exposed elsewhere and, for this reason, we decided to make our programs flexible by leaving the grammar review optional.

The explanations are as brief as they can effectively be made. The lessons are intended to be a review of significant points only, and we felt no need to attempt to be comprehensive in our treatment. On the other hand, every attempt is made in these grammar presentations to provide a unique learning experience. It was anticipated that many students using CLEF might well do so for the purposes of remediation. Many might therefore have found traditional media--teacher, textbook, language lab, etc.--inadequate for their needs. Because of its array of technical features -- colour, screen management, cursor control, simulated movement -- the colour micro offered original possibilities for the presentation of information. In the grammar section we have tried to use the computer screen as a visually-dynamic medium, combining textual explanations with graphic demonstrations. By using colour-coded genders--blue for masculine, red for feminine--we can visually reinforce an explanation of the concept of gender. By extension, the notion of the agreement of adjectives can be visually represented by having adjectives change colour depending on the gender of the noun with which they appear. Syntax and proper positioning of words are suited to visual representation by means of simulated movement. Adjectives can be made to appear at one side or other of the screen and drop into their proper position either before or after a noun located in the centre of the screen. Contraction of à or de with an article can be shown by moving words together in a relatively rapid, though perceptible, sequence: for example, de les, deles, dels, des. The feminine possessive ma cannot be used before école. With simulated movement, we can make the two words bang together and simulate the appearance of rejection by combining this with the micro's blinking feature. Visual calm is restored only by the appearance of mon.

In addition, the use of coloured graphics and simulated movement have been useful in providing animated pictures of the ideas involved, for example, to illustrate the prepositions of place - on, under, etc. or expressions of quantity - a lot, too much, etc. More complex grammatical concepts, such as possession or verb tenses, are also elucidated by the accompaniment of graphic animation.

### (ii) Vocabulary

A potential problem for any CALI program that is intended to be text-independent is that of users being unfamiliar with the vocabulary used in the lessons. Such problems, if left unattended, can quickly lead to frustration. In CLEF, therefore, each grammar exercise is prefaced by a presentation of all the major lexical items that the students may meet in the subsequent exercise. Firstly the items are presented individually, nouns being colour-coded according to gender, and the student can either request an English translation, or press return to move onto the next item. When all the vocabulary has been presented, those items for which the student requested a translation appear on the screen, listed according to grammatical category. Finally, the student has the option of being drilled on the items he did not know. He is given the French word and asked to type in the English equivalent.

In so far as it was feasible, the vocabulary in any given lesson has been made to relate to a specific theme. In this way the lessons serve also as a vehicle for the acquisition of vocabulary. For example, lessons one to four revolve around l'éducation, later ones around la maison, les vêtements, les aliments, les sports, etc.

Lastly, as a final aid to students, a dictionary bank of all words used in the exercises is available on request. The student need only type in the unfamiliar word or expression to receive a translation. This feature helps ensure that students are able to use the lessons on their own. As well, it means that the lessons can be used in conjunction with any grammar text and in a variety of educational systems.

### (iii) Exercises

Each lesson includes from two to four drill-and-practice exercises on the specific grammatical point. The student may choose to do any or all of the exercises in any order he wishes. These exercises adhere to a number of principles which have enabled us to meet our objectives of creating materials that are at once pedagogically sound and well adapted to the medium of the colour micro.

(a) Firstly, in each lesson the exercises show a progression from those intended to perfect mechanical skills, to those which test the mastery of those skills in contexts that are semantically and gramatically meaningful. In some initial exercises the student is faced with no greater linguistic problem than the completion of an essentially mechanical operation: matching articles to nouns, subject pronouns to verb forms, the insertion of an appropriate adjectival form, etc.

The subsequent exercises offer a greater challenge. The student is faced with a series of contextualized sentences in which, in order to perform a required operation, he must take stock of the semantic and grammatical context provided in the stimulus. In the following example:

Dans une semaine il y a ______ jours.

the student must understand the sentence before replying correctly. In this second example:

Non, la neige n'est pas noire, elle est ________.

the student must also be cognizant of the grammatical context, and therefore make the necessary agreement. In some exercises it has been possible to incorporate all of the stimulus sentences into a semantically meaningful continuum, relating to a specific theme or situation. (Holmes, 1980).

(b) Most pedagogues agree that, particularly in the early stages of learning when a concept may offer considerable challenge to a student, it is sound practice to develop recognition skills first and then lead the student by degrees towards the ultimate goal of active mastery. We have developed our exercise formats with this progression in mind. At the same time, attempts have been made to elicit student responses that evolve from mechancial manipulation of the concept towards more subtle applications. Often, for example, a series of four exercises will begin with a multiple choice exercise, followed by a fill-in-the-blank, which is in turn followed by exercises requiring the student to illustrate a mastery of the correct form as well as an awareness of the proper contexts in which it should appear. Progression and variety serve several useful pedagogical and psychological ends. In his book Derek Rountree stresses the student's need to feel he is advancing. In order to do this, he suggests that the media should provide the student with a variety of contexts in which to try out his skills and, as well, progressively more difficult applications of his knowledge. (Rowntree, 1974)

(c) These considerations point out one of the basic principles of good lesson design which is, in order to keep user motivation high, lesson format should be extremely varied. Again the need is particularly acute when dealing with educational technology. Lesson designers must disguise the fact that the learner is faced with an intrinsically undynamic machine that is incapable of spontaneity. Unfortunately, many CAL programs have been sorely lacking in innovation.

In the CLEF exercises we have attempted to engage and retain the user's attention by adopting a number of different strategies. In addition to varying the number of items in a multiple-choice format, we physically position the list of possible answers in different locations on the screen. The fill-in-the blank stimulus sometimes includes two-sentence dialogues in which the words roll on the screen at the rate of spoken speech. In another format, animated graphics are used as part of the question. Yet another possibility is to have the answer from a previous question move to a new location where it will become part of the subsequent question.

The student may be required to input a letter, a number, or a whole word or phrase. Where desirable, independent cursor control will permit the student to insert his answer(s) into different parts of the sentence.

Similarly, once the input has been analyzed, feedback is given in different ways. Incorrect input may be signalled by a textual message, the flashing, change of colour, capitalization or underlining of the correct answer. The incorrect version may be rectified by having the correct version appear on the screen, or by modifying the incorrect input.

(d) These latter points constitute error analysis and feedback. This is an area in which we have striven to fully utilize the potential of the computer as a remedial tool. (Kidd, 1980) Some CAL programs have offered only rudimentary analysis of input. We wished to provide the student with as much detailed and pertinent information on his errors as possible.

Our system combinesdirect-match identification of specific errors with string-matching techniques that enable us to perform a grammatical anaylsis of certain types of input. A bank of anticipated errors helps to detect common errors. The analysis of strings of characters can identify incorrect grammatical entities within a word. Further assistance is provided by a character analysis of input in order to detect minor deviations such as orthographical errors and missing spaces. In most CLEF lessons deviant characters are underlined and appropriate error messages are given.

In some instances, of course, the input is so removed from the correct response that pertinent error analysis is not possible. At such times the student is provided instead with information that will guide him towards the correct answer.

## 2.4 The CLEF Lessons: Level II

Thus far we have mentioned only CLEF level I. The level II lessons, designed for use by intermediate students of French, are similar in most essential respects to their level I counterparts. At the intermediate level, where the majority of students are perhaps better equipped to assimilate pedagogical materials through the traditional combination of textbook and teacher, we provide grammatical information only for remediation purposes, that is when the student makes an incorrect response. At that time he is given some assistance, and often asked if he wishes more. In this respect, the provision of grammatical information is more closely geared to remedial need than in level I. A second difference is that there is no presentation of vocabulary preceding each exercise. At this level of competence, problems of lexis are minimized and it was felt that a vocabulary presentation feature would have little effect on the student's ability to complete the exercise. Thirdly, French is used throughout the level II lesson.

## 2.5 Results

To date, all of the projected sixty level I lessons have been written, and approximately half the lessons are available on disk, along with over one third of the level II lessons. Since the first lessons were programmed, they have been given preliminary field-testing in the three participating universities and at a number of high schools in Ontario. A sampling of initial reaction has proven extremely favourable. All due allowance given to any influence of the Hawthorne Effect, student users, virtually without exception, have identified the CLEF lessons as being potentially beneficial in the study of French. The grammatical exercises were said to be clear, helpful and, because of the various features such as colour and movement, interesting. Many felt that the actual exercises were the most useful part of the lessons, mainly because of the reinforcement and feedback. All users stated that they would use the lessons on a regular basis, should more become available.

Teacher reaction has been similarly positive, though we realize that those secondary and post-secondary teachers who requested the use of the CLEF lessons were already tempermentally and intellectually well-disposed towards the use of educational technology in language education. Nevertheless, even those who had never used the computer felt that the lessons would have a useful role to play in developing reading and writing skills. A significant comment made by those who used the lessons was the fact that many students who normally appeared to be passive and disinterested in class showed greater motivation and became actively involved in the learning of French when they used the computer.

The financial support from the Ontario Ministry of Colleges and Universities and the University of Western Ontario, as well as reactions of students and teachers to our programs, have given us every encouragement to continue in our research. Administrators at the secondary school level have thus far limited themselves to verbal rather than economic support. They perhaps await the results of the more extensive field-testing. Our future plans, upon completion of the programming of the entire series of lessons, include the undertaking of a full-scale, empirical evaluation of the CLEF lessons in particular, and also, of the general effectiveness of CAL as an aid to second-language learning.

# REFERENCES

Adams, E. N. Morrison, H. W. & Reddy, J.M. Conversation with a computer as a technique of language instruction. In R. C. Atkinson & H. A. Wilson (Eds.) Computer-Assisted Instruction: a book of readings. New York: Academic Press, 1966.

Allen, J. R. ELSE at Dartmouth: an experiment in computer-aided instruction in French. French Review, 1971 (April) 46 (5), 902-912.

Brahan, J. W. & Westrom, M. L. NATAL-74: first results. Proceedings of the 1978 ADCIS conference, Dallas, March 1-4, 1978, 202-211.

Holmes, G. A contextualized vocabulary learning drill for French. Computers and the Humanities, 1980, 14, 105-111.

Kidd, M. E. & Holmes, G. CAI for language remediation at the university level. Proceedings of the Fourth Western Educational Computing Conference, San Diego, November 20-21, 1980, 141-147.

Kopstein, F. F. Why CAI must fail. Educational Technology Review (Series No. 9: the computer and education). Englewood Cliffs: Educational Technology Publications, 1973.

Marty, F. & Myers, M. K. Computerized instruction in second language acquisition. Studies in Language Learning, 1975 (Fall), 1 (1), 132-152.

McEwen, N. Computer-assisted instruction in second language learning: an Alberta project. Canadian Modern Language Review, 1977 (January), 33 (3), 333-343.

Meredith, J. C. The CAI author/instructor: an introduction and guide to the independent preparation of computer-administerable instructional materials in the conversational mode. Englewood Cliffs: Educational Technology Publications, 1971.

Olmsted, H. H. Two models of computer-based drill: teaching Russian with APL. Slavic and East European Journal, 1975 (Spring), 19 (1), 11-29.

Rowntree, D. Educational technology in curriculum development. London: Harper and Row, 1974.

Smith, P. D. A computer-assisted instructional review of basic Spanish. System, 1976 (October), 4 (3), 182-190.

Suppes, P. & Van Campen, J. Stanford program in computer-assisted instruction for the period April 1, 1968 to June 30, 1968, progress report. (ERIC Document 034393).

Taylor, H. F. DECO/TUCO: a tutorial approach to elementary German instruction. Foreign Language Annals, 1978 (September), 12 (4), 289-291.

Terry, R. M. Students work with MONIQUE and learn French. Foreign Language Annals, 1977 (April), 10 (2), 191-197.

Villeneuve, R. Enseignement de la grammaire assisté par ordinateur (EGAPO). Proceedings of the Third Canadian Symposium on Instructional Technology, Vancouver, February 27-29, 1980, 255-262.

# Cooperative Courseware Production:
# A Solution to the Problems of CAI?

Glyn Holmes
University of Western Ontario

Donna Mydlarski
University of Calgary

Dana Paramskas
University of Guelph

## Abstract

There have been a number of factors that explain the failure of CAI to live up to its early potential, including the lack of quality courseware packages. The Universities of Western Ontario, Guelph and Calgary believe that they have found one possible solution to many of the problems by cooperating on courseware production. The paper is a description of the cooperative system and its various stages.

## 1. The Problems

In the early years of CAI a good number of practitioners glimpsed the potential of the new medium and made excited prognostications about its impact on education. We were told that by the end of the 1970's CAI would be an integral part of all levels of the educational system (Loughary, 1966), that a new type of teacher would emerge (Gerard, 1967), that teachers and computers would soon be partners (Meredith, 1971). On the other hand there were some dissenting voices which warned that the general hooplah had a slightly hollow ring to it (Kopstein, 1970; Anastasio and Morgan, 1972).

As we survey the current scene we might be forgiven for wondering who was right, the enthusiasts or the scentics. The potential for CAI is still there, but the actual progress made is not sufficient to convince some observers that there has in fact been any progress at all over the last fifteen years (Bozeman, 1981). Certainly there are computers in the schools, but not all educators are quite sure what to do with them (Magarell, 1979), and even in institutions where they are integrated into the curriculum it is mostly in the context of courses on computer literacy rather than CAI.

It would be possible to cite many factors that have combined to produce this state of affairs. Some, such as the natural conservatism of teachers and the educational system (Oettinger, 1969), are external to CAI itself. A perhaps more critical factor is the fact that those of us involved in CAI have not been able to prove that CAI is not a "white elephant" (Fielden and Pearson, 1978), that CAI is indeed worth the investment of time, energy and money. To our mind this problem centres not so much around the problem of hardware costs--micros have put computers within the reach of all educational institutions--as around the creation and dissemination of quality courseware.

It has been said that the lack of readymade courseware packages is the main reason why CAI has not succeeded (Frenzel, 1980). Some packages are available, and we are in fact witnessing a growth in the number of computer software publishers (Curtin, 1981). Unfortunately, many of the packages are insubstantial in terms of lesson time, some programs are of dubious quality (Roblyer, 1981), whilst others may have a noticeable methodological bias that may render them unsuitable for adoption in other institutions. One solution to the courseware problem would be for institutions to write their own courseware, but there are few incentives for individuals to engage in this activity. It is well known that coursewriting brings one few institutional rewards. Funds to hire programming personnel are hard to obtain from wary, sceptical administrators, for there is no guarantee that costs will be amortized over a period of years, or through widespread use. Most CAI materials are in fact known to remain within the institution where they were created, either because the author does not feel they are polished enough, or because the computer system for which they were created inhibits transportability, or because of the lack of convenient, established channels of dissemination (Hunter et alia, 1975).

## 2. Cooperation: A Solution

The institutions represented in this paper believe that they have found a solution to some of the problems outlined above: cooperative courseware production. For over four years we have been engaged in a cooperative endeavour that pools the expertise of high school and university teachers of French, a teacher of adults, an educational psychologist and computer scientists. These individuals represent different schools, faculties, universities and, at the upper end of the spectrum, provinces of Canada. The aim of this cooperation has been the production and testing of CAI courseware for French, that is usable not only at the participating institutions but also in any introductory French course designed for anglophones.

### 2.1 The Evolution of the Project

Cooperation can be planned down to the finest detail before it is implemented. This was certainly not so in our case. It is as much by accident as by design that we are here today. Only a small part of the cooperative venture was the result of before-the-fact planning; the rest was an evolutionary process that came about through the promptings of politicians, luck, circumstance, and the eventual realization that something good was being formed.

The element of planning took place at Western in 1977. The fledgling Faculty of Arts CAI facility, with its four terminals attached to the university's mainframe, had been given a three-year lease of life during which it was expected to do something of note. There was a very limited number of faculty members interested in CAI, only one very modest vocabulary program for student use, and a part-time student programmer (whose salary could not exceed $3,000). The task seemed daunting. Seeking assistance and collaboration seemed to be a logical step. If persons with similar interests from different faculties could be brought together, the venture might have access to more expertise and budgets, and at the same time acquire a higher profile. Faculty members from the Faculties of Education and Part-Time and Continuing Education subsequently joined the group. The next step then seemed to be to see whether local high school teachers might be willing to join in the creation of materials that would be of benefit to both levels of education. Five teachers came to the original meeting; three have been working with us for almost four years.

It was realized that the university's mainframe could not adequately serve students in London high schools. Microcomputers seemed the obvious answer, though there was the inevitable stumbling block of money. An application for funds for hardware and personnel was refused by Western, though a simultaneous request sent to the Ontario Ministry of Colleges and Universities was fortunately more successful. We were however told that we would likely receive only 50% of the request, unless we collaborated with other universities. As a result the CLEF inter-university collaborative project was initiated. It was agreed that Waterloo would provide systems and software expertise, whilst Western and Guelph would concentrate on the writing of courseware. Interestingly, less than twelve months later, Western provided a generous subsidy for the CLEF project.

The FLIC project at Calgary was born in late 1979, and represented a collaborative effort between faculty members in the Faculties of Humanities and Education. No CAI facility existed for modern languages, though there was an established Computer Applications Unit in the Faculty of Education, where considerable expertise could be tapped. A survey of externally available materials revealed the existence of the CLEF project, and this led to a visit to the facilities at Western. It was at this point that the seeds of inter-provincial cooperation were sown. It was decided that, if at all possible, Calgary would utilize similar hardware to that adopted by the three Ontario universities, and that we would share experiences already gained with respect to pedagogical strategies and programming techniques. The possibility of inter-provincial cooperation was written into the Calgary request for funds, and the

Secretary of State, through the Alberta Department of Advanced Education responded by funding the project for the last three years.

The inter-provincial component of the cooperative effort thus initially constituted two separate projects, which agreed to a common basis of hardware and pedagogical strategies. It was not long, however, before it became apparent that there were strong reasons for standardizing the CLEF and FLIC lessons, thereby effecting a merger of sorts. Standardization involved even closer collaboration on courseware production, and it is the workings of this collaboration that we shall now describe.

### 2.2 The Cooperative System: How It Works

It would be all but impossible to define the contributions made by individuals in the production of our CAI materials. The lessons may be characterized as mosaics that blend the efforts of up to a dozen people or more. The pedagogical materials, and the computer programs into which they are incorporated, are literally shuffled across Canada, to be viewed, reviewed, tested, modified and updated.

The courseware production system comprises several integrated, interdependent levels (Fig. 1).

#### 2.2.1 General coordination

Perhaps the most important component of the cooperative system is the overall lesson coordination. At the very outset of the CLEF project it became obvious that certain ground rules and conventions would have to be adopted if the lessons were to have any appearance of consistency. Those of us principally involved in the design of the courseware had to agree upon the general pedagogical content of the lessons, their format, the type and level of vocabulary to be used, and the language skills they were intended to develop. Such concerns are of course endemic to any kind of coursewriting. Other questions were however specific to CAI development: the coordinated use of color, movement, timing, screen management, types of error analysis and feedback, help features, use of graphics etc. In order to design such a system, multi-level cooperation is crucial. Because the lessons are destined for use by students of varying ages, from youngster to adult, care must also be taken that the features incorporated in them--for example, in the area of grammar presentations, aid facilities etc--will be appropriate for all levels. Only the collective efforts of a group of instructors with combined experience at the various levels of teaching and with the various teaching methods can produce courseware that will promote effective learning among a broad spectrum of users.

#### 2.2.2 Lesson writing

As shown in Figure 1, Guelph provides both Western and Calgary with the written version of the grammar presentations that begin each lesson. In this way the types of information selected and the approaches used in the presentation of material acquire some consistency. At their respective ends, Calgary and Western write a series of graded exercises to drill the student on the grammar point. At Calgary the exercises and the formulation of error analysis are done by a single content specialist, whereas at Western it involves the team of high school and university teachers outlined earlier. This latter group meets every two weeks, and each member is made responsible for the linguistic materials in a set of exercises. The materials are then offered to the group for comment and criticism. This is in fact only the first of many stages in the production process when materials are scrutinized and modified by others.

Subsequently a full-time program coordinator revises the exercises on the basis of the suggestions, and she formulates specific error analysis and feedback.

#### 2.2.3 Lesson coordination

The next stage involves the coordination by Western and Calgary of the Guelph grammar review, and the exercises. In a perhaps unusual situaton for academics, particularly humanists, each institution allows the others to make changes in the materials it has authored. This implies a large amount of faith in the other members of the project, though any major changes are always preceded by consultation. Changes are in fact often necessary if the grammar and exercises are to complement each other. Ideas contained in the one part may involve revision of ideas in the other.

#### 2.2.4 Scriptwriting

Once the linguistic contents have been established the subsequent step is to ensure that the CAI program produced for the computer is actually what the content specialists intended. This requires close collaboration between coursewriter and programmer. The coursewriter composes a detailed script incorporating the linguistic contents, error analysis, feedback and all details relating to the manipulation of these materials on the screen. It is the programmer's job to translate this information into a BASIC program. The script serves a vital function: it compels the coursewriter to clarify his own thoughts. It is all too easy to formulate a vague idea in a vague manner, and leave the precise interpretation of meaning to the receiver. The script ensures the clear exposition of ideas, and thereby the efficient cooperation between coursewriter and programmer.

### 2.2.5 Programming

Our programmers have shown themselves to be very good at doing what they are asked to do. On the other hand, they are invited to offer criticism and suggestions on the ways in which the computer can mediate the linguistic materials. They, after all, are the experts in inventing ways of exploiting the computer's potential. Programmers have often made valuable suggestions for streamlining lessons, for example, by creating subroutines, random-access comments, etc. In addition to their area of expertise they have also, on occasion, made very useful suggestions concerning the visual design as seen from the user's viewpoint, as opposed to that of the pedagogue.

If there is cooperation between coursewriter and programmer, there is also cooperation between programmers at different universities. Western, for example, had written a series of data insertion programs and a spelling check routine which was sent to the other members of the project. Similarly, Guelph has made available a number of help feature routines and a quit routine which have recently been incorporated into all lessons.

### 2.2.6 Debugging and revision

Once the computer programs have been written there is the inevitable need to detect the errors and debug. A fresh eye is needed, and at Calgary and Western the programs are reviewed by the content specialists. It is very rare that the specialist is totally happy with the lesson. The lesson in its original script form often seems different from its actual realization on the computer screen. At other times points clearly need revision. The programmer's sanity must be borne in mind when requiring changes though, as we shall see later, this is the only time that an individual programmer is likely to make substantial revisions to his own programs.

### 2.2.7 Testing

It is of course very important to subject our materials to the scrutiny of potential users, both students and teachers. In so doing we seek not only confirmation of our faith in the quality of the lessons, but also criticism which might lead to improvements. The testing is done in all three universities, though not necessarily in the same way or in the same type of situation. In Guelph and Calgary testing has been done under experimental classroom conditions with students enrolled in the regular introductory French courses. The experimental testing is complemented by less formal assessment whenever the opportunity arises. At Guelph and Calgary, for example, the university's Open House allows us to gauge the reactions of a range of people, from grade-schoolers to senior citizens. In Guelph's Language Laboratories the microcomputers are placed among VTRs and other equipment for library-type audio and audio-visual materials. This design allows the coursewriter to observe actual student use of the programs, as well as to obtain, through the lab assistants, an accurate record of technical difficulties and unanticipated errors.

For its part, Western is more involved in field-testing the materials at locations other than the university campus. The secondary school teacher included in the Western group provide an ideal means of testing in the schools. In addition, a machine is reserved for field-testing at outside locations. The microcomputer and materials have been shipped to a number of high schools throughout Ontario, as well as to places as far away as Northern Quebec to be used with native peoples.

### 2.2.8 Revision

The testing performed to date has given us insights into user reaction that have led to revisions in the CLEF/FLIC lessons. Several changes have been made at Guelph to enhance learner convenience. An in-exercise translation feature was added to allow the user to ask for the meaning of a word during the actual lesson: an "out" option allowing users to quit at any time in the grammar reviews or exercises: a grammar reference feature providing a reminder of the main points touched on during the grammar review. In addition to making these improvements, it was agreed to use a standard format for the CLEF and FLIC lessons, and this has meant reviewing all lessons and making appropriate changes in graphics, screen management, colors and the like.

Though the three groups must agree on any significant changes such as these, the actual revisions and updating are performed by Guelph. There are two principal reasons for this. Firstly, a programmer, or even a whole group, at an institution can react adversely to making seemingly endless changes to his (their) own programs. Secondly, the process of standardization is facilitated if it is done in one location.

2.2.9 Final review

When the revisions and updating are completed by Guelph, the lessons are returned to the institutions where they were originally programmed. By this point the lessons are virtually in their final form. However, even at this stage there are occasional minor bugs to be ironed out. This is not so much a comment on the programming, but on the complexity of some of the lessons, and the number of checks that we run on them. It is well known that insufficient debugging is one reason why many CAI materials have been regarded as substandard. With this final check the cooperative sequence of courseware production is completed, and the mark I version of the lessons are ready for implementation.

2.3 Reflections on the First Four Years

The cooperative venture has evolved over the past four years. During this time we have been able to reflect on the advantages and disadvantages of the various levels of cooperation, and on the degree to which they may have helped solve some of the problems outlined at the beginning of the paper.

It would be difficult to assess the extent to which cooperation has decreased the costs of CAI development; indeed it might be possible to argue that the costs have not been diminished. The majority of the funds for the project have derived from government sources, and we therefore cannot claim that individual institutions have benefited in terms of a diminished investment from regular budgetary sources. On the other hand the fact that funds have come from outside sources is of decided benefit to the institutions, and it is certain that the fact of cooperation has been instrumental in obtaining at least part of the funds, particularly as far as the Ontario universities are concerned.

The picture is still clouded when viewed from a different angle. There is no doubt that cooperation on lesson production has avoided duplication of effort, the constant byproduct of insularity. It is on the other hand highly likely that more effort has been expended on the production of the CLEF/FLIC lessons than would have been the case had the teams been working independently. It is only if we accept that similar lessons of similar length and complexity would have been created that we can talk of the cost-benefits of cooperation.

If we consider the question of the potential use of the materials, the cooperative project offers considerable advantages which could be construed as financial. Our CAI materials have a considerably enlarged group of actual users with the involvement of three universities and one Guelph and two London high schools. In addition, the number of potential users is significantly increased by two factors. Firstly, if three universities can agree on common courseware, there is every likelihood that others will also find them suitable. Secondly, given the major input by secondary school teachers, the materials should be relevant at this level too. Thirdly, software routines and error-analysis strategies, once designed and debugged for French are easily transferable to other types of content, such as other Indo-European languages.

In the long term it seems very probable that the cooperative project will prove to have been a sound investment. However the advantages of cooperation are perhaps not best evaluated through the narrow prism of cost-benefits. Perhaps more important returns derive from the improved quality of the CLEF/FLIC courseware. The quality is improved in a number of ways. The materials are pedagogically sound in that they represent the pooled expertise of several French instructors at different educational levels, as is true for any well designed courseware. There is also a large amount of variety in the materials and the ways in which they are treated. There is little doubt that many heads can conceive more imaginative ideas than one. At times the desire to avoid the boring repetition of lesson formats and methods of presentation have proved a stumbling block to the rapid development of lessons; however, the potential positive effects of originality on user motivation outweigh the extra investment of time. The widespread testing also ensured better feedback on the lessons and more significant improvements being made. Furthermore, the lessons are more reliable because of the extensive debugging.

An intangible benefit that should nevertheless not be overlooked is the simple fact of cooperating with similar-minded people at other institutions. "Colleagueship," as one of us put it, has its own rewards: the enriching exchange of ideas and sharing of initiatives: the creative pressure produced by the knowledge that one's own work affects that of others; the realization that the end-product of this collective effort surpasses anything one might have created alone. To some scientists such feelings may not be new; to humanists, who have a deserved reputation for working in splendid isolation, the experience has proved unusually rewarding.

This does not of course mean to imply that cooperation has not had its attendant problems. Coordination, particularly at a distance, is not always easy, and the problems can multiply with the number of persons involved. The problems have been minimized by frequent meetings, often at conferences such as this. Disagreement can sometimes occur, and they can only be solved by a sincere willingness to compromise. The compromises have been, if numerous, relatively small. Differences in materials and methodologies are sometimes not as radical as might at first be believed. One final complaint might be voiced by our respective budget heads: the cost of long-distance telephone calls.

## The Future

Like all similar projects, we are faced with the problem of continued funding. The Ontario Ministry of Colleges and Universities has stated that this is the final year of funding for the CLEF project. Concerted attempts are being made to obtain regular funding from the university, and with a substantial set of CAI materials that can be implemented into introductory and intermediate French courses, there is the possibility that such funding will be forthcoming. At both Western and Guelph the university has in fact already provided some funds, though not on a committed, on-going basis.

It may of course be possible to tap other government or private sources, though any application would be viewed in a less favorable light if it were clearly an attempt to supplement the regular operating budget. A new project might be more successful, and feedback on field-testing has revealed a vital area for future collaboration: the translation of the CLEF/FLIC lessons to run on the popular micros.

•

## References

Anastasio, E., Morgan, J.S. Factors Inhibiting the Use of Computers in Instruction. New Jersey: EDUCOM, 1972.

Bozeman, W.C. "Computers in the Schools: Introduction to Special Issue." Educational Technology, 21, 10 (1981), 13-14.

Curtin, C. "Publication Practices for Microcomputer Programs." Journal of Computer-Based Instruction, 7, 4 (1981), 123-29.

Fielden, J., Pearson, P.K. The Cost of Learning with Computers. London: Council for Educational Technology, 1978.

Frenzel, L. "The Personal Computer--Last Chance for CAI?" Byte, 5, 7 (1980), 86-96.

Gerard, R.W. (ed). Computers and Education. New York: McGraw-Hill, 1967.

Hunter, B., Kastner, M.L., Rubin, M.L., Seidel, R.J. Learning Alternatives in US education: Where Student and Computer Meet. Englewood Cliffs, N.J.: Educational Technology, 1975.

Kopstein, F. "Why CAI Must Fail." Educational Technology, 10, 3 (1970), 51-53.

Loughary, J.W. (ed). Man-Machine Systems in Education. New York: Harper and Row, 1966.

Magarrell, J. "Use of Technology for Teaching Said Lagging, Despite Colleges' Big Spending on Computers." Chronicle of Higher Education, 19, 8 (1979), 1, 5.

Meredith. J.C. The CAI Author/Instructor. Englewood Cliffs, N.J.: Educational Technology, 1971.

Oettinger, A.G., Marks, S. Run, Computer, Run: The Mythology of Educational Innovation. Cambridge, Mass.: Harvard University, 1969.

Roblyer, M.D. "When is it 'Good Courseware'? Problems in Developing Standards for Microcomputer Courseware." Educational Technology, 21, 10 (1981), 47-54.

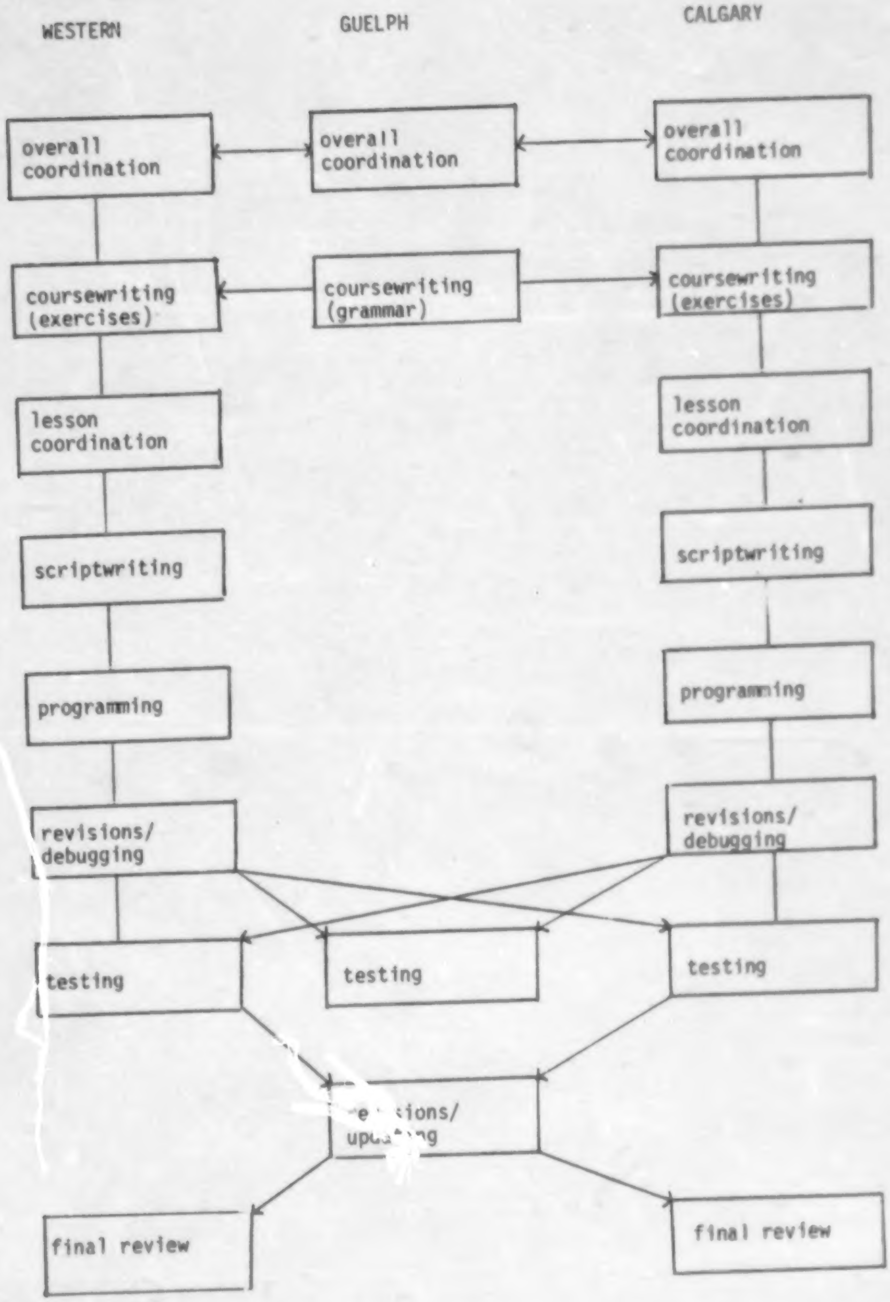

WESTERN
GUELPH
CALGARY
overall coordination
overall coordination
overall coordination
coursewriting (exercises)
coursewriting (grammar)
coursewriting (exercises)
lesson coordination
lesson coordination
scriptwriting
scriptwriting
programming
programming
revisions/ debugging
revisions/ debugging
testing
testing
testing
revisions/ updating
final review
final review

Figure 1

The Cooperative Coursewriting Structure

# THE FLIC PROJECT : MICROS AND MINIS

DONNA MYDLARSKI, ANN BREBNER AND H.J.HALLWORTH

THE UNIVERSITY OF CALGARY

## ABSTRACT

In the University of Calgary project on French Language Instruction by Computer (FLIC), CAI programs have been developed to provide drill and practice in written French. These may be used, at university or school level, following class instruction on a given topic, and free the teacher to concentrate on oral work. The programs are implemented on a microcomputer using a colour monitor. They are also available on a mini-computer, with the microcomputer being used as a terminal. In the latter case, a management system on the mini-computer keeps a record of student responses and progress. This allows courseware authors to monitor programs so that formative evaluation becomes an on-going process. Teachers may obtain regular reports on individual students, or groups of students, so that instruction may be tailored to the needs of the learners. A controlled study of the FLIC programs and their integration into the instruction offered by the Department of Romance Studies is at present being carried out.

## FLIC LESSONS: CONTENT

The project on French Language Instruction by Computer (FLIC) at the University of Calgary is concerned with the development of programs which are used following class instruction on a given topic. By providing drill and practice in written skills relating to grammar, syntax, and the mechanics of language, they free the teacher to concentrate on oral work.

The programs are written in BASIC for a Compucolor II microcomputer. They are also implemented in DEC BASIC PLUS on a PDP 11/70 mini-computer, to be delivered via a Compucolor II used as a terminal with the larger machine operating a management system.

Since lessons for a number of initial topics had been produced in the CLEF project, lesson development for the FLIC project began approximately half way into what is regarded as the first level of a larger continuum. A list of topics, as given in Table 1, was established in consultation with faculty members teaching beginners' French. These topics are ordered in a manner commonly found in textbooks; however, the lessons are self-contained and text independent. Moreover, the sequence is not regarded as fixed, and teachers may accommodate individual student or class needs by rearranging the units. For example, a student having trouble with pronoun objects could be referred to Lessons 2 and 6, and then 5 and 7.

Colour is used as a means of coding masculine and feminine nouns, and number and gender of verbs, also for highlighting specific words. Movement of words and simple graphics are also used. Directions appear in French because students who have reached this point in the continuum are considered sufficiently competent to read instruction in the target language.

An attempt is made to involve students in the exercises by asking questions of personal interest. The lesson on use of the verbs aimer (to like) and détester(to hate) with direct pronoun objects, for example, asks "Do you like spinach?" This may be answered by "I like it" or "I hate it". When using the elusive pronoun *en*, the student is asked "How many sisters do you have?" and "How many cars does your friend have?" The program on the future tense asks questions such as, "In 1988, will you be studying at the university?"; "In 1988, will you be living in Canada?"; "In 1988, will you have children?"

In some programs, for example those concerned with syntax, sentence elements are randomly generated and students are required to unscramble them. In drilling verb tenses, a paragraph is displayed in the present tense, verbs are deleted, and the student is required to complete the paragraph in the past or future tense.

Analysis of student errors is followed by appropropriate cues and prompts. In the case of multi-word input, this may involve complex procedures. Following the question "Do you like spinach?" acceptable answers are "I like it", "I do not like it", "I hate it", or "I do not hate it". The analysis is, of course, complicated by the fact that, in French, use of the negative adds two words to the sentence.

Positive reinforcement is graded on two levels. Superlatives such as *formidable* or *fantastique* are used for correct answers. A milder comment such as *juste* or *bien* is given when the correct answer has required a prompt.

At the end of the exercise, the student's score is displayed. Questions that were missed reappear in a review, which enables the student to improve his/her performance. However, in order to avoid "over-kill", only one attempt is allowed in the review.

Table 1

Lessons in the FLIC Continuum

1. le passé compose des verbes en - *er*
2. les pronoms objets directs
3. l'accord du participe passé des verbes conjugués avec *avoir*
4. le passé composé avec l'auxiliaire *être*
5. le pronom *y*
6. les pronoms objets indirects
7. le pronom *en*
8. le futur simple

## THE MANAGEMENT SYSTEM

When the CAI programs are used from the PDP 11/70, with the Compucolor II as a terminal, the management system maintains a record of student responses, and thereby permits the courseware designer to monitor and modify the lessons as desired. Formative evaluation may then become an ongoing process.

A CAI lesson is divided into three or four exercises, each having up to ten questions to which the student is required to respond. A student may repeat an exercise as many times as desired, and on each repetition is presented with the review, described earlier, which provides opportunity for a further attempt at any question answered incorrectly. The management system records, for each student, the number of times each exercise is repeated; the total time, in seconds, spent on the exercise; the number of times each question was presented; and whether the final response to the question was correct.

For each lesson there are up to six anticipated errors, determined by the courseware authors on the basis of their previous experience in teaching the content. For each student, a record is also kept of the number of times each anticipated error was made.

Progress reports for individual students, or for groups of students, may be obtained at any time. Also, summary data may be obtained for any group of students, regarding performance on any lesson, exercise, or question, also regarding the frequency of anticipated errors.

## FORMATIVE EVALUATION AND ERROR ANALYSIS

When the CAI lessons were used with a pilot group of 45 university students taking a beginning French course, it was found that the mean time for an exercise ranged from 1½ minutes to 18 minutes. In cases where the mean time appeared unduly long or short, the design of the exercise was examined to determine whether such a time might indeed be reasonable. For example, a test in which error messages have been suppressed will be much shorter than an exercise in which elaborate diagnosis of errors occurs. The unscrambling of the elements of a sentence tends to take a longer time as it tests a global knowledge of the language rather than discrete points. In the case of a seemingly excessive amount of time, it may be that the student is being asked to read a lot of text or to make lengthy inputs. Or the exercise may be too difficult.

Mean scores on all exercises were high. For example, in the three exercises for Lesson 6, they were 99.5 percent, 92.4 percent and 98.3 percent.

This suggests the questions were at an appropriate level for these students. However, high scores were sometimes the result of frequent repetitions. One question in the second exercise of Lesson 6, for example, was repeated on average almost seven times per student. An implication is that learning by CAI was highly motivating: not infrequently, students repeated an exercise until they obtained a perfect score.

Exercises repeated most frequently were those identified by the authors as requiring a higher level of competence. These included, for example, exercises in which students had to make use of two concepts, such as direct and indirect object, or expressions using both *avoir* and *être*.

Some anticipated errors occurred frequently, and it was apparent that there was a need for the remedial messages which followed them. Two such messages, for example, were related to gender: *adresse est feminin*; and use of the auxiliary verb: *vous voulez l'auxiliaire être*. On the other hand, it was found that some remedial messages were never used by this group of students.

The formative evaluation resulted in a number of modifications to the lessons. One exercise was identified containing vocabulary which had not been thoroughly assimilated due to infrequent exposure to certain idioms such as, for example, *avoir honte*. The situation is being corrected by the introduction of additional drill prior to this exercise. In other exercises, the same error was made repeatedly. It was apparent that some students did not know whether certain common words require a direct or indirect object; for example, *le téléphoner* was used instead of *lui téléphoner*. Similarly, students frequently misplaced the pronoun object when used with a semi-auxiliary verb such as *pouvoir*, *devoir*, or *vouloir*. In both cases, a remedial loop will provide the necessary additional practice before the student proceeds with the exercise.

In summary, the management system is invaluable if the FLIC project is to take full advantage of the computer (Hallworth and Brebner, 1980). It enables courseware authors and instructors to monitor programs and to identify areas in need of further development. Instructors may also make use of the diagnostic reports for individual students and courses.

## CONCLUSIONS

In general, the FLIC programs are working satisfactorily, with learning difficulties the same as those normally encountered in the teaching of French as a second language. They may now be used instead of drill and practice exercises which ordinarily take place during class time under the supervision of the instructors. Alternatively, they may be used to provide drill and practice which is additional to that available during lesson periods.

If we may assume that these programs will produce the same learning effects as are found in other subject areas, students who use the programs instead of engaging in drill and practice during lesson time will make the usual improvements in less time. On the other hand, students who use the programs for supplementary work, in addition to the regular class work, will reach higher levels of achievement (Thomas, 1979).

However, the major problem in either case is to integrate use of the programs into the on-going instruction offered by a university department which has well established procedures for teaching beginning French. Given that universities are highly conservative, the problem should not be taken lightly. Use of the programs instead of the usual instruction involves logistical problems in ensuring that all students have access to a limited number of microcomputers for an adequate amount of time, also in ensuring that students do in fact use their allotted time effectively. Use of the programs as supplementary instruction involves not only logistical difficulties but also morale problems: with conflicting demands upon their time, students tend to neglect course activities which are "voluntary" in the sense that they do not account for part of the course grade.

With cooperation from colleagues within the Department of Romance Studies, such problems are now being resolved so that a controlled study of the FLIC programs may be undertaken. In the first instance, this will involve use of CAI time instead of scheduled class time. However, it is later intended to invite students to use the programs on a voluntary basis, in an attempt to establish familiarity with CAI in a Faculty of Humanities where it is not ordinarily available. Preliminary results suggest that such experiments will prove very successful in the formation of positive student attitude.

Hallworth, H. J. and Brebner, Ann. Computer Assisted Instruction in Schools. Edmonton: Alberta Education, 1980. ERIC document ED 200 187.

Thomas, D. G. The effectiveness of computer-assisted instruction in secondary schools. Association for Educational Data Systems Journal, Spring, 1979, 12 (3), 103-116.

ACKNOWLEDGEMENT

The project reported in this paper was supported by the Department of the Secretary of State, Ottawa, whose assistance is gratefully acknowledged.

# DESIGN AND IMPLEMENTATION OF A MICRO-BASED CAI COURSE FOR A MICROFICHE PROCESSOR

James A. Cook

Standard Oil Company of California

## ABSTRACT

During 1981, the Computer Services Department of the Standard Oil Company of California and a practicum group from California Polytechnic State University worked together to develop a CAI course. Criteria were developed for judging a course's suitability for CAI presentation. A number of candidate lecture/demonstration courses taught at Standard Oil Company of California were considered, and the KOMSTAR Formatting Course met the criteria best. The objective of this course was, and is, to teach people the necessary control statements to produce microfiche using the KODAK KOMSTAR 200 Microimage Processor.

In this paper the development of the KOMSTAR CAI Course is discussed from the selection of subject to the evaluation stage. Points of particular interest include a description of why this course is suitable for CAI development and how the development progressed through course requirement analysis, course specification and software design.

## INTRODUCTION

The KODAK KOMSTAR 200 Microimage Processor is a high-speed printing device utilizing laser and photographic technology to produce printed output on microfiche. To use this device efficiently one must learn how to prepare parameter statements. These parameter statements contain information concerning system and job setup requirements, data input requirements, and microimage output characteristics. These parameter statements are also called parameter cards, a hold-over from the days of punched card input. In the past at Standard Oil Company of California (Socal), the users of this system have been taught how to code these statements, or cards, by attending a one day lecture/demonstration class. The class was designed for people with limited data processing experience.

In December of 1980, a practicum group[1] from California Polytechnic State University was formed in cooperation with Standard Oil Company of California. This group, after examining a number of courses taught in-house at Socal, decided that the KOMSTAR Formatting Course was the best candidate for conversion to a CAI mode. This decision was based on a number of reasons:

1. The KOMSTAR course was highly modularized. There were ten separate parameter statements which were taught along with some additional statements in a special software package called the Parameter Management Executive (PME). The majority of these statements were logically independent of each other, thus, a student could take a lesson on a particular statement without regard to the other statements.
2. Unless there was a desperate need, there had to be a minimum number of students signed up for the class before it was taught. The course was not in demand by a large number of people and was

---

[1] The author wishes to gratefully acknowledge the participation of Ray Dean, Mike Gregory, Karl Kunz, and Lori Leonard who, under the supervision of Dr. Emile Attala of California Polytechnic State University, contributed to the successful completion of the CAI KOMSTAR Course.

offered, on the average, about every 8 weeks. CAI could alleviate this problem by making it possible to provide this course on an individual, as-needed basis.

3. The class under TAI (Traditionally Administered Instruction) had more information than some individuals needed. With a well-planned CAI modular course an individual could choose to learn only about the parameters that related to his or her needs.
4. There had been no refresher course for students who wanted to review how to use the KOMSTAR parameters. CAI would allow a student to review material selectively.
5. KOMSTAR coding was difficult for some people and easy for others. CAI's self-paced attribute would allow the student to proceed at his or her own pace and not at the pace determined by an instructor.
6. In TAI, some people had not asked questions for fear of appearing ignorant. This inhibition caused problems later in the course. CAI instruction could alleviate this problem by presenting the course content to the student at his or her individually chosen pace and by giving the student many opportunities to review the course material.

## IMPLEMENTATION REQUIREMENTS

Since Socal was working with certain CAI equipment, the course had to be designed to be used at a learning station that contains an Apple II Plus microcomputer, one color monitor, one 5¼ inch disk drive, and a videotape player.

During the course development, the necessity of a workbook was clearly recognized. The development of a workbook for the course was considered to be essential for at least two reasons: the resolution of the computer graphics was too low to depict accurately the various coding forms and microfiche formats, and the workbook would make it more convenient for the student to maintain a sense of sequence throughout the course.

There was one major software requirement. BASIC had to be the language for courseware coding because it was the highest level language available for the microcomputer learning stations of Socal at the time. This forced choice did lead to some serious problems.

## COURSEWARE DESIGN AND IMPLEMENTATION

After deciding on which course to convert to CAI mode, and becoming aware of the hardware and software requirements, the question of courseware design and implementation was addressed. The course development process was broken down into three major stages: course requirement analysis, course specification, and software design. The course requirement analysis in turn was broken down into three topics: student profile, course content requirements, and course structure requirements.

The student profile was developed by examining the students who normally take the KOMSTAR course. These students are employees of Socal whose day-to-day responsibilities include either skill in formatting microfiche or supervision of people so skilled. These students are not restricted to one predominant age group or sex and come from a wide spectrum of positions and locations. Job titles such as "Programmer/Analyst" and "Systems Analyst" are typical, but many non-programmers take the course also. The students are required to have some working knowledge of IBM's Job Control Language. Some students have no microfiche-formatting background while others have obtained some knowledge by reading manuals and simply wish to learn more about certain parameters.

The course content requirements depend to some degree upon the course objectives. This course, at Socal, had two broad training objectives:

1. The student is to attain mastery of the microfiche-formatting capabilities of the Kodak KOMSTAR 200 Microimage Processor available through the company-wide time-sharing system.
2. The student is to be made aware of the advantages and disadvantages of microfiche in general and particularly the cost savings that can result from the intelligent use of microfiche.

The course was not expected to cover all of the powerful capabilities of the KOMSTAR or all significant aspects of microfiche. Rather, the course was designed to cover only those microfiche-formatting features relevant to the Socal work environment. Emphasis was placed on specific formatting features that were commonly used.

In this case the course structure requirements mainly evolved from considerations of the type of course presentation material that was to be used. The course had to be structured to take advantage of the computer as a vehicle for self-paced, programmed instruction. The practicum group decided that the minimum structural requirements imposed on the course were:

1. Instructional material must be broken into small units.
2. Each such small unit must terminate with feedback on progress in the form of questions.
3. The student need not demonstrate mastery of each unit's questions before being allowed to advance to new material.
4. Units should be grouped into lessons.
5. Lessons would be preceded by pre-tests, which would determine if the student needed to take the lesson.
6. Lessons would be terminated by post-tests, which would evaluate the student's mastery of the lessons.
7. Videotape should and could be used if it was appropriate.

For ease of access and control, the course would be "menu-driven" and CRT screen advancement would be student-controlled by single key responses. Using menus, the student could always select from clearly explicit options and would be free to access the lessons in any sequence or omit certain lessons according to his or her needs.

## COURSE SPECIFICATION

The course specification was divided into three topics: course structure, learning objectives, and testing policy. The course structure was developed after studying the topics to be taught. The decision was made that the course should consist of 18 "lessons". Each lesson would consist of a sequence of teaching "units". Each unit would require multiple CRT screens and one or more workbook pages.

Each lesson would present a logically self-contained topic. Clearly defined learning objectives would be presented in the workbook at the start of each lesson. Lessons would be numbered 1 through 18 and vary in size. The student would be expected to complete each lesson in 20 to 40 minutes.

The teaching units within each lesson would develop the lesson topic in an incremental manner. There would generally be five or fewer progress questions at the end of each unit. These would be used to provide feedback as well as build the student's confidence. Units would vary in size, but the student could expect to complete a unit in 5 to 15 minutes.

Most of the CRT screens would require the student to reference a page in the workbook. The convention used to reference the workbook would be as follows: if a workbook page number appears in a conspicuous green box in the upper right corner of the CRT screen, the student is to reference that workbook page while the screen is displayed. If no green box appears in the upper right corner of the screen, no workbook page need be referenced.

Each of the 18 lessons would be accessed through the KOMSTAR MENU. After logging onto the course, the student would be presented with a menu of lesson choices which would include an introductory videotape segment.

Each lesson would have only one entry point, the beginning of the lesson. If the student selected a lesson from the KOMSTAR MENU, he or she would enter the lesson and would be immediately presented with three choices before taking the lesson. They could return to the menu, take the lesson, or take the pre-test to determine if it was necessary to proceed with the lesson.

Upon completion of a lesson, the student would not be rushed into the lesson's post-test, but would be given the option to review individual units as long as they wanted. The student would have to decide when the amount of review was sufficient and he or she was ready to take the lesson test.

The content of the instructional material was specified by the learning objectives. These are the same objectives that were addressed by the lecture/demonstration course. It is enough to say that each lesson and unit has at least one listed learning objective.

The testing policy developed included pre and post-tests and progress questions. Immediately upon selecting a lesson, one of the options available to the student would be to take the lesson pre-test. The pre-test would consist of approximately 10 questions which would pertain to the lesson's learning objectives. If a score of 70 percent or higher was obtained on the pre-test, the lesson material was to be considered mastered and the student advised that he or she need not take the lesson. The student would be advised of the score obtained on the pre-test and it would be recorded on the student's progress record.

The post-test at the end of each lesson would be very similar to the lesson pre-test. To exit a lesson after entering the instructional sequence, the student would have to take the lesson's post-test. Upon completion of the post-test, the student would be informed of the score obtained and advised whether or not to re-take the lesson.

It is important to note that tests (as opposed to progress questions) would be designed solely to evaluate mastery. Test questions should be posed with a randomly-sequenced degree of difficulty. The student would not be informed which questions were answered incorrectly, nor would any feedback be given (except the score and advice as to whether the lesson should be re-taken).

Progress questions, on the other hand, would be designed to assist in the learning process. Progress questions, unlike test questions, would be posed with gradually increasing difficulty, and feedback provided for each question. In fact, different wrong answers could result in different feedback. Progress questions also would not be scored.

## SOFTWARE DESIGN

The code structure of each lesson should mirror the logical decomposition of lessons into units and units into screens. Each screen would be coded as a subroutine. A unit would be a set of subroutine calls to appropriate screens. Since screens were to be coded as subroutines, they would be conveniently shared by multiple units when necessary. The screen driver for each unit would also be coded as a subroutine. Thus a set of subroutine calls at the beginning of the lesson's code would act to call the lesson's units as needed. This code could be presented as self-documenting in a top-down manner. For example, given a specific screen ID code, it would be easy to randomly access any screen. Rather than comb sequentially through a listing of screens, the source code reader need only start at the unit driver, proceed to the appropriate screen driver, and immedaitely locate the address of the desired screen.

## RESULTS AND CONCLUSIONS

In October of 1981, the CAI KOMSTAR Course was delivered to Socal. The course was in BASIC code and each lesson took up one floppy disk. The course also included a 7 minute opening video segment. This video segment was particularly impressive in that the student was given an overview of the microfiche printing process that would not otherwise be possible. The KOMSTAR Microimage Processor was shown and the normal path of microfiche creation at Socal was traced. Very few of the students who took the course in the TAI mode had the opportunity to be exposed to this knowledge.

The course as delivered consisted of approximately 40,000 lines of BASIC code. We consider this to be a large programming project for educational software development.

The microfiche course is presently in the evaluation stage, and feedback from the users (students) is being used to continue polishing and fine-tuning the course. The basic design appears to be reasonably sound and the course objectives are being realized. The 40,000 lines of BASIC code are very difficult to maintain and clearly demonstrate the necessity of using some other language to produce educational software for a microcomputer. The cost savings and convenience of this type of instruction have been demonstrated. People are obtaining the needed training in an interesting, motivating, time efficient manner. In general, for one individual to receive the same training using the TAI mode, the cost was in excess of $300.00. The CAI training costs are a fraction of that amount. The CAI concept appears to work well with this course at Socal and more courses are being planned and developed using this concept.

# AN APPLE COMPUTER AS A TERMINAL FOR A UNIX BASED EDUCATIONAL NETWORK

Gareth R. Shearman

COMPUTER USING EDUCATORS OF BRITISH COLUMBIA

## ABSTRACT

The feasibility of establishing a Province wide educational data network in British Columbia is under active investigation. The Computer Using Educators of B.C. are studying the potential of such a network for their members. We have been experimenting with connecting Apple microcomputers to a mini computer running under the UNIX operating system. Given the appropriate hardware and software configuration, few serious difficulties were encountered. The problems that did arise were, or potentially could be, solved by various strategies. The benefits of access to such a system were found to far outweigh the deficiencies. We have experimented with or envision developing a range of services including program sharing, a "help line", conferencing and general member intercommunication.

The feasibility of establishing a Province wide educational data network in British Columbia is under active investigation. Representatives of a number of educational organizations are involved in the investigation. These include the three universities and the Knowledge Network. The tentative plans for "Edinet", as it has been dubbed, call for an initial stage which would link the two universities in Greater Vancouver, The University of British Columbia and Simon Fraser, with the teaching hospitals and the B.C. Institute of Technology. Later stages of the network would add the University of Victoria and the regional community colleges. One of the organizations which is very interested in these developments is The Computer Using Educators of British Columbia (CUEBC), which is an association of teachers formed under the aegis of the B.C. Teachers' Federation, the professional teachers' organization in B.C.

In September, 1981 CUEBC began a continuing study of the potential for its members of such a network. With the generous assistance of the University of Victoria's departments of Computing Science and Creative Writing, CUEBC has been given access to a research oriented 11/44 which is running version 7 of the UNIX operating system. UNIX is the popular multi-user operating system developed by the Bell labs. It is widely used in educational institutions and has been proposed as a potential operating system for the future generation of 16 bit microcomputers. Any member of CUEBC with access to a modem equipped microcomputer may participate in the study. As many of our members have Apples we have been gaining experience with interfacing this type of micro to UNIX.

In general, we have encountered few difficulties with using an Apple as a UNIX terminal. There are, of course, some serious hardware deficiencies in the Apple system. These include the lack of a complete ASCII character set, a 40 column screen and an awkward, hybrid method of moving the cursor around the screen. The hardware and software options available to the individual user will have a significant bearing on the end result. The configuration available to the author includes the following items:

Apple II plus with a language card
Two Apple disk drives
Epson MX 80f/t printer
DC Hayes Micromodem II
Modem Magic ver. 2.0 (software terminal driver)
Magic Window (word processing software)

Having both the DC Hayes modem and the Modem Magic package available from the start meant that the author experienced little difficulty in his initial attempts to access UNIX. The Hayes is a direct connect device which provides for dialing the phone directly from the keyboard. Entering a few carriage return characters during the sign on procedure was enough to clear any garbage characters and make UNIX accept the Apple as a valid terminal type. The Apple's 40 character line length would have caused serious problems with line truncation. The "Modem Magic" software package, however, provides for automatic wrap around so this potential problem was successfully by-passed. At this point the author found himself free to wander around gleefully exploring a brand new multi-user operating system on a mini computer - an order of magnitude increase in capability over the Apple itself.

The UNIX C.A.L. program "learn" provided a handy means of acquiring at least the rudimentary skills. The mail and message functions were tried and found to work smoothly. While proceeding through the first of the mini-courses within "learn" the first snag was encountered. The Apple lacks a left hand square bracket character, which is used by UNIX for a number of purposes. This problem proved to be only a temporary nuisance, however, for as soon as Modem Magic version 2 was obtained the left hand square bracket was provided by a control character. More problems soon presented themselves, however. Later in the "learn" sequences and in conversations with other UNIX users the utility of the back slash, tilde and straight line characters became evident. The lack of these characters prevents the Apple/UNIX user from readily employing a number of powerful UNIX concepts such as the use of "pipelines". There are at least two possible solutions to this problem. Version three of Modem Magic, it is rumored, will contain the missing characters. There is another possibility which would make use of some of the features of the UNIX operating system. Each UNIX user has quite a lot of control over his own personal interface to the system, his envelope or "shell". It should therefore be possible to develop a software Apple interface which would substitute other, unused characters for the missing ones.

UNIX systems include a very powerful screen oriented editing program known as "vi". The problem of allowing a user with an Apple terminal to use this program is, at this writing, being addressed by our local guru - (every university based installation has such an individual, quite often in the person of a graduate student). For the moment, we are restricted to using the standard UNIX "edit" program. An alternative, particularly useful where long distance or computing charges intervene, is to use some form of word-processing package for the Apple computer and then transfer the result to UNIX. One of the problems with this approach can be found in the way in which the various word processors store their files. Applewriter, for example, stores a binary image of the file which when transferred to a UNIX system results in an unusable and incomprehensible file. Some of these programs, including "Magic Window" which the author is using, allow a text version of the file to be saved. The Modem Magic program, in its turn, can then access the file and transfer it to the UNIX system where it arrives as a perfectly readable file of text. There certainly are some drawbacks to the current arrangements but, with some patches and a bit of ingenuity most can be overcome.

In spite of some of the above deficiencies, access to a UNIX based network has provided CUEBC with some important pluses. In a network environment it is easy for two users to work together on a project. Messages and ideas may be shared using the facilities of "mail" or perhaps a computer conferencing system. In addition, files may be shared so that two or more individuals may access an interim copy of a report or book, enter their contributions and leave it for the next person. Besides the obvious advantages of being able to pass messages back and forth and co-operate on reports and projects we have found another significant use. The Modem Magic package includes utility programs for translating Applesoft and Integer BASIC programs, as well as binary files, to text files. These files can then be transferred by the modem. The program also includes the capability of entering "accept mode" where all incoming information from the remote system is retained in memory. The amount of material that can be stored is substantial, about 28k. We have, therefore, set up a directory where the text versions of public domain programs can be stored and accessed by any of our members that may wish to use them. All such an individual has to do is to load the text version into his Apple's memory and, with the use of the appropriate routines, convert the program back to a usable form. Once anything has been stored in the Apple's memory by using the "accept mode" feature of Modem Magic it can be saved to disk or dumped to a printer. The above sequence doesn't overcome the problem of unreadable binary files mentioned earlier, the idea is to have the material on UNIX in readable form.

The access to expertise afforded by a network is also proving to be an important consideration. Early minor annoyances such as lists of names or files which scrolled by in a single column too fast to read were quickly reformatted for us. The on-line UNIX manual's nasty habit of demanding a c/r for every line displayed was also soon overcome. The establishment of a permanent "hot line" for dealing with questions and problems is a definite possibility once more of CUEBC's members gain access to the network. We are also interested in the potential for submitting articles for our journal in a form which can be quickly retrieved and processed by our editor.

At present we are just beginning to explore the potential. The network is only in a pilot form, a far cry from the Province-wide educational data network which is envisioned. We are just beginning to understand even the rudiments of the UNIX operating system. Even at this early stage, however, some important conclusions can be identified. With minimal effort, a microcomputer can be configured to function as a terminal for a networking system. The benefits which can be obtained from access to such a system encourage us to promote its rapid further development.

# Computer Literacy--Keeping the Options Open

Annalee Elman - American Institutes for Research

## ABSTRACT

As education moves to including computers in the school curriculum, questions arise as to what this curriculum should consist of. Several definitions of computer literacy which have been proposed are presented and discussed. Educators are urged to keep the options open, and to allow local objectives to evolve as more equipment becomes available and teacher expertise increases. END

Whenever I find myself in the company of teachers-turned-computer-buffs at conferences, I am impressed by the atmosphere that their excitement generates and which amounts to almost evangelical fervor. Indeed, I sometimes find it a challenge to temper my own enthusiasm with the caution and moderation that befits a Ph.D. One sure-fire method for bringing myself down to earth is to put my energies and mind to work on an analysis of some of the problems that many of us seem to face in winning others over to our particular and individual vision of the way things should be. The current debate over computer literacy has that kind of sobering effect on me, because of the questions it raises and the consequences it implies.

As I surveyed the literature on the subject in preparing this paper, I was struck by the range of issues that the various authors have proposed for inclusion in a definition of computer literacy. For quick comparison, a non-representative sampling of definitions and their sources is summarized in Table 1. For convenience, each entry in the table is checked against a list of topics and concepts that have been proposed as goals and objectives for the United States Schools in Europe's Darmstadt Career Center, as reported by Kauffman (1976). The Kauffman list was used because it is an early list and because of its breadth, not because I believe it to be "right" in any absolute sense. The sources summarized on the table are identified in the bibliography.

Fitting individual conceptions into a framework, no matter how broad, loses detail. The following discussion restores some of the context of the originals. My apologies go to any authors who feel their representations have been slighted.

The National Council of Supervisors of Mathematics position paper (NCSM, 1978) is more of a policy statement than a specification of objectives, and should be viewed in that light. The NCSM endorses and reinforces the position of the National Council of Teachers of Mathematics (NCTM, 1978) in calling for a recognition in our curriculum of the importance and pervasiveness of computer technology in all facets of society. The NCSM position in 1978 emphasized awareness, so as to dispel the "mystique" surrounding computers. The NCTM position in 1978 was to emphasize "first-hand experiences with both the capabilities and the limitations of computers through contemporary applications." A more recent (1980) NCTM recommendation identifies "computer literacy" with "familiarity with the role and impact of the computer" and urges inclusion of this component in the general education of every student and as a requirement for all math teachers. Experience with computer programming is treated as a separate topic.

Reaching back still further into the history of the definition controversy, we find a definition of computer literacy proposed by the Committee on Computer

Table 1: Computer Literacy Objectives from Selected Sources

| | CCE 1972 | Holzman & Glaser 1977 | NCSM 1978 | Johnson et al. (MECC) 1980 | Luehrmann 1981 | NCTM 1978, 1980 | Cupertino USD 1981 |
|---|---|---|---|---|---|---|---|
| I. Understand functions and use of components of a computer system | X | | | | 10% | | |
| 1. Definitions of terms | Should understand what computers can and cannot do | | | X | | | X |
| 2. Ability to use handheld calculator | | | | | | | |
| 3. Describe how a computer works | | | | X<br>including I/O devices | | | X |
| 4. Operate all hardware | | | | | | | X |
| 5. Perform computations in a variety of numbering systems | | | | | | | |
| 6. Use the operating system features | | | | | | | |
| 7. Run a packaged program | X | | | | | | X |
| II. Use systematic problem solving techniques to develop computer solutions | X | X | | | 80% | | |
| 1. List sequence of steps (algorithm) | Discuss algorithms, how they are represented by programs, and how executed | Stress acquisition of programming skills as a medium for introducing math and logic concepts | | X<br>Much attention devoted to following logic, debugging and developing and modifying algorithms | | | X |
| 2. Draw a Flow Chart | | | | | | | X |
| 3. Write programs that run as planned | | | | | | | X |
| 4. Document programs | | | | | | | |
| 5. Program in at least two languages | | | | | | | |
| 6. Load a compiler | | | | | | | |

Table 1 (continued)

| | CCE 1972 | Holzman & Glazer 1977 | NCSM 1978 | Johnson et al. (MECC) 1980 | Luehrmann 1981 | NCTM 1978, 1980 | Cupertino USD 1981 |
|---|---|---|---|---|---|---|---|
| III. Analyze the role of Computer Technology in society | X | | | | 10% | | |
| 1. Describe ways in which computer is being used in government, communications, military, business | Stresses present uses and impact on the individual | | X<br>Also, limitations | X | | X | X |
| 2. Describe how advances have affected family, school | | | X | X | | X | X |
| 3. Describe possible negative effects | | | | X | | | X |
| 4. Analyze an innovation for possible uses and misuses | | | | | | | |
| 5. Describe historical development | | | | | | | X |
| 6. Speculate about the future uses of computer technology | | | | | | | |
| 7. Identify and explore related occupations | | | X | X | | | X |
| 8. Analyze differing viewpoints on uses | | | | | | | |

Education (1972) that places heavy emphasis on understanding, limiting hands-on experience to the use of packaged programs. The comment "where time and equipment becomes available," included in their definition, clearly dates this interpretation and explains the restrictions placed on it.

The Holzman and Glaser article (1977) is cited here as an example ot the consequences of adopting one definition over another. It describes a study in which six sixth graders were taught programming skills using LOGO and FOCAL. The title of the paper, which describes this study, is Developing Computer Literacy in Children. The authors evidently subscribe to the definition of literacy as "doing computing." Knowledge about computers and computing is acquired incidentally, if at all.

The Johnson et al. (1980) reference is to a set of objectives developed by the Minnesota Educational Computing Consortium (MECC), which forms the basis for the Minnesota Computer Literacy and Awareness Assessment Test developed by MECC in 1979. This test, a paper-and-pencil instrument, imposes severe limitations on the objectives that can be measured in a true-false or multiple-choice format. Luehrmann (1981) rightly criticized these objectives because they define computer literacy in terms of what was being done in schools prior to 1979, confuse computer awareness with computer literacy, and place too much emphasis on knowing about computing at the expense of knowing how to compute. Luehrmann argues that the MECC objectives are interesting as a snapshot of computer literacy in schools in 1978-79 but that they tell us precious little about what a curriculum of the 80s ought to contain.

The response of Johnson et al. to Luehrmann's criticism raises some valid points regarding the need for what they term "ordinary people" to explore and consider the social and economic impact of computers on their lives. However, I submit that to devote entire courses to such a consideration in the absence of hands-on, practical experience with computing is to run the risk of creating opinion without basis in reality. We call such opinions prejudices.

The computer vocabulary and definitions a child needs to have can be incorporated into a curriculum with an emphasis on doing. Vocabulary and definitions are useful only insofar as they are used. They will be used when the student needs to request a floppy disk or a teacher points out that the reason a program failed to execute was because a variable name contained too many characters or the execution required more core storage than was available. Vocabulary is valuable in that it enables communication on matters in which individuals are personally involved; learning definitions is valuable in that it enables accurate, precise communication on these matters.

To postpone hands-on experience to later in the term or, as happens too often, later in the schooling life cycle, is to deny children an opportunity to learn something as soon as they are able to function at it successfully. I'd like to illustrate this point with the growing number of experiences, formal and informal, documented and otherwise, that indicate that young children, younger than many would have believed possible, are perfectly capable of learning to program. Given the proper "environment," children as young as three can learn to operate computers, insert disks or modules, and engage in interactive CAI. Primary school children are doing their own programming, using such "child-friendly" languages as LOGO. Furthermore, our preconceptions about which children are able to learn programming are likewise being challenged by experimental programs at MIT, in New York City, and elsewhere. White or black, high, average, or low IQ, young, younger, or youngest--empirical evidence that exists shows clearly that the only limitations are the child's own interests, motivation, and willingness to learn. Since this is the case, any school-imposed restrictions on who is to have access to the computer, and especially the exclusion of children from programming instruction on the grounds of low ability, youth, or poor performance in math, can only be viewed as a form of social control that is indefensible on pedagogic grounds.

The challenge to teachers is to educate ourselves, to keep open, and not to impose or project our own limitations on the children we teach. Consider that computers have been in existence only 35 years. New developments, new applications, new possibilities are coming into being almost daily. To remain intellectually honest, we must not let our beliefs become etched in stone as soon as they are formed. As the field evolves, so must we. We should review and renew our computer curriculum as often as possible. This means not merely taking what is as "given" and "refining" it, as Johnson et al. (1980) term it, but subjecting our objectives annually to an audit that can be likened to "zero-based budgeting". We must ask each year the questions, "Based on new developments in the field, what should remain of last year's curriculum?" and "What new components should be added?"

My remarks about controlling the impulse to institutionalize computer literacy standards prematurely applies to standardizing our testing procedures in this area as well. I see that paper-and-pencil standardized, multiple-choice testing is as out of place in a computer curriculum as it is in physical education. Hands-on learning can only be measured satisfactorily by performance--the ability to write an algorithm, construct an accurate flowchart, debug a program, write one that works. Quizzes, either oral or written, can be used to identify and clear up errors of fact, but should not be mistaken for measures of competency.

There is another form of testing, now out of vogue, that may be applicable in upper grades and after sufficient exposure to hands-on computing, namely essay tests. There are serious issues and implications surrounding the spread of computing power--issues such as the invasion of privacy, computer crime, the ethics of restricting access, the concept of information as a resource and its implications, and the consequences to a society of robots, working at home, and "prosumerism" (a Toffler term) to name just a few. These are issues best explored in dialogue and discussion, and best presented for thoughtful assessment in writing.

The challenge to educators is to recognize the amazing power and range of these small boxes as a new medium for problem solving. The final entry in Table 1 (Cupertino, 1981) is a set of objectives used in a Northern California school district. It is a good example of "what is." In this program, students become familiar with the microcomputer through "canned" programs in grades K-3, learn to read and draw flowcharts and create simple programs in grades 4-6, and are introduced to algorithms in grades 7-8. Word processing is also introduced in middle (or junior high) school. The "doing" component constitutes 16 out of 69 objectives, or less than 25% of the total. Furthermore, hands-on programming, introduced in grades 4-6, is not treated in grades 7-8 beyond "translating mathematical relations and functions into a computer program." These limitations imposed on the curriculum are a recognition of local limitations in teacher expertise, availability of hardware, and probably district policy. It is unlikely that they reflect the creators' ideas of what could be, given reduction or elimination of any of the above stated limitations.

What's missing in the above curriculum? For starters, programming is not introduced until grade 4. Then instruction in programming is limited to BASIC and to solving math problems. Applications to "word processing" are not introduced until grade 7. No opportunity is provided in these objectives for children to create stories, program self-quizzes, or explore the range of special features (sound, color, graphics) on their computer and learn to incorporate them into their programs. The implication here is that programming is too hard for younger children and should be limited to older students who are good in math or have learned to type. The other children, too young or too stupid, will learn sterile facts dressed up as social studies, history, or science and to spout back teacher opinion as if it were the truth. The computer will become merely an excuse for parading out the same old values in new garb.

I'll conclude with a definition of computer literacy proposed by Elizabeth Glass, Consultant for Computer Technology of the Connecticut State Department of Education. She proposes the following set as minimal competencies:

All students with the exception of the most severely (educationally) handicapped should be able to:

- o Use a variety of pre-programmed computer programs. (This includes selecting the materials needed for a specific purpose, putting the program into the computer, and executing the program.)
- o Write a program to instruct the computer to do what the student wants it to do.
- o Understand the economic and social impact of computers on individuals and on society.
- o Use the ideas of programming and computer applications in developing solutions to problems. (Glass, 1981)

Let us recognize that computer education in our elementary and secondary schools in in a trial-and-error stage. We must not be afraid to experiment, dare, or dream on a grand scale. Nothing great has ever been accomplished by thinking small.

## References

Anderson, R. E., Klassen, D. L., & Johnson, D. C. In defense of a comprehensive view of computer literacy--A reply to Luehrmann. Mathematics Teacher, December 1981, 687-690.

Committee on Computer Education. Recommendations regarding computers in high school. Washington, D.C.: Conference Board of the Mathematical Sciences, 1972.

Cupertino Unified School District. Computer Literacy Curriculum. The Computing Teacher, September 1981, 29-34.

Glass, Elizabeth M. Computer literacy: What is it? ACES Exchanges. New Haven, Connecticut: December 1981.

Johnson, D. C., Anderson, R. E., Hansen, T. P., & Klassen, D. L. Computer literacy--What is it? Mathematics Teacher, December 1981, 681-686.

Kauffman, D. T., Jr. Teaching the future: A guide to future-oriented education. Palm Springs, Calif.: ETC Publications, 1976.

Luehrmann, A. Computer literacy--What should it be? Mathematics Teacher, December 1981, 681-686.

National Council of Supervisors of Mathematics. Position paper on basic skills. Mathematics Teacher, February 1978, 147-52.

National Council of Teachers of Mathematics. Position paper on basic skills. Mathematics Teacher, February 1978, 147-52.

National Council of Teachers of Matheematics. An agenda for action. Reston, VA: Author, April 1980.

# COORDINATION AND EVALUATION OF VIDEODISC AND MICROCOMPUTER PROGRAMS ACROSS 45 SITES

Donald H. McLaughlin - American Institutes for Research

ABSTRACT

This paper describes the videodisc/microcomputer (ViM) network, a collection of 45 elementary and secondary schools across the country that intend to use microcomputers as videodisc controllers to teach basic skills. Included are the rationale and objectives for the network, the process of implementation, and specific plans for courseware evaluation.

## RATIONALE

Technological development requires feedback, and the marriage of videodisc information storage with microcomputer control is certainly no exception. This combination opens new possibilities for individually adaptive instruction in basic skills, offering opportunities for many wrong decisions as well as right ones. How long should video sequences be? How should computer-graphics be used with videodisc-generated displays? Can one videodisc be designed for use by a variety of computer programs? Although we can use logic to address these questions, the answers, as well as new, overlooked questions, will come only from experience. This project is designed to accumulate experience and to analyze it to provide guidance for future development of videodisc/microcomputer (ViM) technology.

There can be no question that this technology brings new capabilities to the classroom. With a microcomputer alone, one can generate a few dozen high resolution pictures from information stored on a floppy disk, while one side of a laser videodisc holds over 50,000 individually addressable frames. One can imagine whole courses stored on a single videodisc, lectures sharing space with visual simulation feedback that will be used for exercises run under control of a computer program. The program itself may be packed densely into a few frames of the videodisc.

There is a distinct drawback of ViM technology in its present form, however. Unlike writing a computer program, which can be done by a teacher in a few hours using new authoring languages, creating a videodisc is "high technology," to be done rarely, and not iteratively as one works out the "bugs." Thus, unlike computer courseware, we are not likely to see local school districts and teachers' colleges developing numerous videodiscs for personal use and experimentation. In the future, "mastering" a disc may become routine, but it is not, using the present technology. This is an especially critical problem in these early developmental stages of the technology, because there are so many wrong ways to implement videodiscs. Therefore, squeezing as much information as possible from each application of ViM in the classroom is crucial for its successful emergence as a practical addition to basic skills teaching methods. Thus, the ViM network project aims to provide assistance to schools that want to try the new technology so that it can be "formatively" evaluated.

## OBJECTIVES

In order to provide the information for making decisions on the development of ViM technology, the U. S. Department of Education set up a project to:

(1) identify schools across the country that have made a commitment to technological improvement of basic skills instruction, where the ViM technology would receive a fair and valid test;

(2) facilitate use of the technology in these schools, through

  (a) establishment of an on-line network, and

  (b) equipment support; and

(3) collect and report information on use of the technology and on the strong and weak points of specific ViM courseware.

## Identification of Schools

Schools selected for this project will play a key role in the future directions of ViM technology, so they have been chosen carefully to provide both a fair test of the technology and a basis for extrapolating conclusions to schools across the country.

The first concern was that the schools have an existing base from which to develop a program of usage of ViM technology. This led to inclusion of the criterion that participating schools should already have possession of a laser videodisc player and a microcomputer of sufficient capacity (48K with disk drive) to run existing educational courseware. For 16 schools in Maryland, Virginia, and the District of Columbia, this equipment had previously been purchased by the Federal government, and for other schools it was a prerequisite for participation.

In order to ensure that the results will be relevant to the teaching of basic skills, the participants have been selected from among elementary and secondary schools. This has excluded a number of colleges and universities that are also exploring this new technology. We feel that the principles and problems of courseware design are sufficiently different for children and adult students to warrant this distinction.

Finally, we have attempted to enhance the generalizability of the results by including schools across the country, with a variety of levels of socio-economic status. On the one hand, we have the inner city schools for which the Federal government has provided the equipment for this project; and at the other extreme, we have affluent and innovative schools that have already amassed extensive experience in using computers for instruction.

## Facilitation of Use of ViM Technology

Because of the significant hurdles to be overcome in putting a new technology to use, the Federal government has taken a leadership role both by funding projects that produce videodiscs for school use, and, in this project, by providing support and incentives to schools to try the technology. We are implementing two broad categories of assistance: providing a medium by which schools can communicate with each other and exchange both programs and insights based on trying these programs out and by providing equipment backup and service. The equipment support extends to purchase of interface cards for schools that do not already have a means of controlling videodisc displays from their computers.

In order to foster communication, we have created a ViM Network, by means of which participants can send messages both to single recipients and to the other participants as a group. Four criteria guided selection of the particular network medium: (1) the medium should be capable of storing and transmitting programs as well as messages; (2) it should have a powerful text editor, so that it can be used to translate programs between dialects of the same language; (3) the cost should be minimal, and also transparent to the participating schools; and (4) the system should be easy to use.

In addition to providing the medium, the project is providing assistance in connecting to the network, including modems to allow microcomputers to act as terminals and, where necessary, telephone extension installation costs. We believe that with this assistance, the only investments required of the schools in order to access the network are (1) the cost of calls to the nearest Telenet node, (2) monthly telephone bills, and (3) time to learn how to use the network.

Equipment support has been a difficult decision, because without limits, the costs can skyrocket. We have restricted outright purchases of equipment to loan to schools to videodisc interface cards (and the previously mentioned modems), and the remainder of our support is in the form of backup equipment to be sent to schools while their equipment failures are remedied, and (within limits) replacement parts. The project staff have taken training courses in servicing the type of microcomputer being used in 90% of the network sites (Apple), and we are assisting some schools directly with equipment maintenance.

In spite of the limited nature of the equipment support, equipment costs comprise over one fifth of the direct expenses included in the contract awarded for coordinating the network and developing and implementing evaluations of courseware.

## Formative Evaluation

The goal of this effort is to find out what the potential of ViM technology is and how best to use it. In order to accomplish this purpose it is necessary to examine specific courseware, both in vitro and in vivo, to determine what works and what doesn't. To carry out this evaluation, we plan to apply three types of observation, and to identify different sources of strengths and weaknesses.

The three types of observation will be:

(1) Content validity, in which we examine the match between the courseware and previously accepted attributes of good courseware, such as appropriate content, appropriate pedagogical principles, clear instructions, informative record-keeping, and motivation enhancers;

(2) Performance changes, in which we embed sufficiently detailed record-keeping within selected software to provide the basis for analysis of components of the learning process; and

(3) Classroom effectiveness, in which we gather teachers' observations of the strengths and weaknesses of the courseware in the contexts of their classrooms.

These observations will be analyzed to identify appropriate avenues for improving basic skill education, within and between the broad categories of (1) changing the characteristics of the videodisc medium, (2) changing the characteristics of the content of videodiscs; (3) changing the characteristics of software written to use videodisc material; and (4) changing the ways in which the use of given courseware is managed.

The results of these analyses will be made available immediately to members of the ViM network and, as quickly as feasible, to curriculum developers at large.

## IMPLEMENTATION

### Identification of Schools

When establishing a network, one does not want to exclude any schools that are carrying out relevant activities and are interested in participating. On the other hand, there are costs associated with each additional participant. Attempts to be all-inclusive might lead to damaging dilution of the assistance the project can give to schools in implementing the new technology, thereby reducing the value of the test of the teschnology. Thus, given the resources available at the Federal level for this project, it was decided that 45 schools should be invited to participate.

When we began; we were not sure how many schools we would find that were prepared to try out videodiac-microcomputer technology. There might be so many that we would have to be highly selective, or there might be so few that we would need to actively recruit schools. We started with the previously selected Washington area schools and called various regions that have been known for their leadership in computer technology in the schools. At the same time we waited for calls from interested schools to the Division of Educational Technology. At present, we are just under our planned limit of 45; and although we have not had to turn away applicants yet, we shall have to in the near future. We have nearly 40 sites participating, from Oregon, California, Utah, North Dakota, Minnesota, Michigan, Indiana, Louisiana, Pennsylvania, New York, Maryland, Virginia, and the District of Columbia; and we hope shortly to add 6 more in different states.

One problem we have had to consider in selecting schools is the variety of hardware being used in the schools. The majority of schools we have been in contact with have Apple computers, and it is clearly more efficient to set up a network with uniform resources. However, we believed that this might inappropriately limit the test of the technology, and so we have included schools with other computers, specifically, Commodore and Tandy computers. We have not yet worked out all the difficulties implied by this decision, however. By implementing the network on a medium with a powerful text editor, we have, hopefully, made it possible to translate BASIC programs, at least, between computers made by different manufacturers.

We do not view the network membership as static. It is possible that one or more schools will find itself unable to partcipate actively, so that the purposes of the project would be better served by replacement with another school. Also, if a school wishes to participate in such a way that no significant additional costs are associated with participation (i.e., the school is responsible for its own equipment and telephone costs), we would not plan to exclude this participation.

### Facilitation of Use of ViM Technology

It was essential to prepare for the on-line mail system as early as possible, to provide a concrete reality to the project. With this electronic mail system, schools could identify a specific network "object." We therefore set up a system with which we are quite familiar, based on the Stanford University computer facility. This required a minimal initial investment, in either time or money, so if there proves to be a better alternative in the future, little will be lost in switching computers.

In fact, however, in addition to being competitive on the cost dimension, this alternative offered several advantages. First, there is the Wylbur text editor, which adds to the mail system the capacity to amend and alter programs and other data sets being stored and transmitted via the network. Second, it gives the project coordinators the capability of controlling the logon sequence so that a questionnaire, for example, can be administered at logon time. Third, by interfacing with telenet and tymnet, it gives us the capability of reversing telephone charges so that schools will not be charged. Finally, it gives us access to detailed records of system usage, while providing the capability for each user to maintain secure files on the system.

At present, we have put the first issue of a periodic newsletter on the Stanford system so that it can be retrieved by network members, and we have also stored a telephone directory of participating schools, a set of BASIC programs, and an Index of data sets available to users. Thus, we expect that by the end of this month, February, 1982, most members of the network will have used the on-line mail service at least once.

To connect to the network, or any other external system, each microcomputer must be equipped with a modem. So far we have focused on the Apple computers in the network and ensured that each has a Novation AppleCAT modem, which includes software for low and high speed memory transmission and switching to local disk access to store and retrieve transmitted information. One of our main problems at present is training the personnel at the participating schools how to use the modem and its associated software to communicate with the network.

Of course, with modems each of the schools can communicate with each other directly, without using the electronic mail system at Stanford, and thus without leaving a trace that can be used to record activity in the network. This provides us with a dilemma, because although the Stanford system can be made convincingly attractive, with its text editing, storage, and wide band broadcasting capabilities, it incurs telephone costs that could be avoided by users who are situated near each other and who call each other directly. Although the telephone costs are transparent to each school, they do in fact subtract from the total resources the project has for assisting the schools and evaluating the technology. Should we encourage schools to call each other directly (e.g., by writing instructions and software to facilitate it) or should we discourage it for the duration of this project?

In addition to the on-line mail system, the project will be assisting participating schools with some equipment needs. We've made some progress in that area, focusing on Apple computers, which are at the large majority of schools. The schools are really in three categories of about equal numbers:

(1) those with significant experience using both microcomputers and videodisc players in instruction, if not together;

(2) those with some experience with microcomputers in instruction who are taking the initiative to acquire videodisc players and otherwise become prepared to participate in the project; and

(3) the schools for which this project is the first real experience with the use of microcomputers in instruction.

The equipment needs of these categories of schools are different, but for each they fall into two types: equipment maintenance and videodisc interfacing.

The project staff have prepared to assist schools in maintenance of their microcomputers by becoming certified as technicians and by acquiring backup computers, spare parts, and diagnostic programs from the Apple computer company.

We have evaluated several alternatives for interfacing Apple computers with videodisc players, and we have found only one that is in a reasonable price range and which can flexibly serve the variety of videodisc players in use in the network. In trying this one with one of the major laser videodisc models (DVA 7820-1), we encountered a hardware problem with the player which was first diagnosed by the manufacturer as a bad card in the player. Later it was rediagnosed by the manufacturer as a design problem in the player that had been cured in later models of the player. This problem has not yet been completely resolved.

## Formative Evaluation

Two data collection forms had to be developed for the evaluation aspects of this project: a ViM Usage Log and a Teacher's Courseware Assessment form. We have developed these and obtained FEDAC and OMB clearance to use them.

The ViM Usage Log is a set of eight multiple choice questions that can be administered twice monthly as a part of the network signon procedure. To implement this, we can set a flag so that the first time each participant signs on, he or she will be asked the questions. After that, the questions are skipped, until we reset the flag.

The ViM Usage Log gathers information on the level of usage of the videodisc and microcomputer equipment for basic skills instruction; on the context of its use (grade level and location); on problems; and on the overall assessment of the contribution of the technology.

The Teacher's Courseware Assessment is one of three methods of data collection for the courseware specific evaluations to be carried out. It consists of seven questions concerning the effectiveness of the courseware, six questions about specific problems that might occur in using the courseware and how they were dealt with, and eight short questions about the context in which the courseware was being used.

The most pressing need at this point is for courseware to evaluate. The Federal government has funded production of two series of videodiscs, Infinity Factory and Music Is, and other discs may result from current projects, but at present we can only evaluate a very few alternatives. Once we have found out what is wrong with these, we must wait for the next generation of videodiscs to find new areas for improving the use of the technology.

## COURSEWARE EVALUATION PLANS

The most useful product of this project will be improved knowledge about how to use ViM technology. The data provided by the participating schools will serve to validate hypotheses about what makes courseware effective in general, and it will provide guidance in the design of videodiscs. Throughout the study we shall try to separate results that indicate a need for improvement of the medium from results that indicate, on the one hand, need for improving the content of videodiscs, and on the other hand, the use of given videodiscs in the classroom. It is essential for this purpose that we not only examine the videodiscs in the lab (content validation) but also record student behavior when interacting with the courseware (process analysis) and teachers' observations of the impact of the courseware in their classrooms (impact evaluation). Our approach is to use all three of these techniques, on selected packages.

The gathering of information will be enhanced if more people become involved in the evaluations, so we plan not only to design the data collection efforts but also to prepare a diskette containing an "Evaluation Design Facilitator" which staff of the participating schools can use to design and carry out their own evaluations of courseware. We hope that the school staff will find this useful both in evaluating courseware prior to purchase and in evaluating courseware that they may develop in house.

The Evaluation Design Facilitator will contain three modes of evaluation:

(1) Questions that the evaluator should address while examining the courseware firsthand;

(2) Instructions for inserting a machine language subroutine that records student behavior into an existing courseware module; and

(3) A questionnaire to administer to teachers whose students use the courseware, with guidelines for interpreting the responses.

## Content Validation

The most inexpensive form of evaluation (other than using somebody else's prior evaluation) is to try the courseware out yourself with a checklist of good points to be matched and bad points to be avoided. Of course there is a danger in using a checklist because you may be tempted merely to add up the plusses and minuses in order to arrive at an overall rating. A more appropriate use of a checklist would relate each plus to an increase in the breadth of situations in which positive results would be obtained and each minus to a restriction of the range of expected positive results.

Similarly, it would be an oversimplification to apply the same evaluation procedure to all courseware, ranging from half-hour exercises to K-8 systems. Realizing this, we plan to evaluate courseware in terms of "sessions", which will become more clearly defined as the project proceeds. Each session can be construed to have three successive components, although one or more may actually be missing for any particular courseware module:

(1) Identify the student and either assess status or retrieve status from records; then select a starting point;

(2) Administer a sequence of "Display/Request" pairs and record responses (the sequence will usually depend on the responses made); and

(3) Score the session and update and report the student's status.

The items to be included in the "checklist" will be similar to those developed by the National Institute of Education and reported by Richard Otte.

- Are the dimensions included in the student records meaningful and appropriate?
- Do the responses called for require the skills to be learned?
- Is the sequencing of material pedagogically sound?
- Are the branches to handle incorrect reponses pedagogically appropriate (do they deal with the problem indicated by the error)?
- Does the videodisc contain a sufficient coverage of alternative paths?
- Are the graphics pleasant but unobtrusive?
- Is the videodisc material used well?
- Is there variety in the material displayed?
- Are displays and response requests free from requirements that don't relate to the lesson's objective?
- Are instructions easy to understand and responses easy to make?
- And, of course, is the program foolproof?

While some of these questions are obvious, others, such as the quality of graphics and the presence of extraneous requirements are not. One important product of this project will be a validation and improvement of such rating dimensions, by comparison with actual performance gains and teacher reactions.

## Process Evaluation

The classroom teacher, who can observe the whole student and evaluate how well the material taught by the ViM technology enhances development, must remain the ultimate source of information on the effectiveness of courseware. Nevertheless, we can learn much about the detail of what works and what doesn't by direct observation of the actual interactions of the student with the system. This is possible because the computer provides us the capability of exact measurement of response times and of recording actual responses made to the stimuli.

Our approach will be to identify particular behavioral objectives of the courseware that can be measured both at the beginning of the interaction and at the end. We will then categorize students into three classes: those who possessed the skill both before and after the session; those who acquired the skill during the session; and those who lacked the skill both before and after the session. By comparing the last two of these groups, we shall be able to isolate the source of problems for the courseware in dealing with the third group.

The first step in the comparison is to identify "subgoals". Did the group that failed to learn achieve these subgoals? If so, then further analysis of the relations between the components and the overall skill are necessary. On the other hand, if not, then we search for the reason the subgoal(s) were not achieved. To address this question, we must categorize responses as indicating or not indicating attentive participation. If the student was responding randomly, for example, we might not expect learning to occur. If the participation appears to have been attentive, on the other hand, then we must look for a problem in the instructional design (such as not requiring practice of the target skill).

If there was no attentive participation, we must separate complete non-participation, which should be counted as a non-test of the courseware, from participation that wanes as a function of either boredom or frustration, brought on by the courseware design.

Gathering data in this form on several dozen students in several schools will provide valuable insights into the range of applicability of particular courseware modules and into the design characteristics that are associated with different levels of applicability.

## Impact Assessment

Finally, as mentioned above, we plan to gather teachers' assessments of courseware modules, using questionnaires containing some 21 items. These items will cover assessment of effectiveness, identification of specific problems, and a description of the context of the courseware usage.

The questions on effectiveness cover student motivation, instructional content, individual diagnosis and assessment, adaptive remediation, and record maintenance. The problems mentioned in the questionnaire include classroom management, hardware and software failures, instructions and documentation, and bridging to other classroom content. One open-ended question is included in which the teacher is asked for any other problems not falling into our preconceived framework. The context questions cover grade level, location, numbers of students simultaneously using it, hours per student, and intended relation to other curriculum components.

It is our hope and expectation that the results of these different evaluation approaches will converge to give us an overall picture of each of the videodisc/microcomputer courseware modules that we select for analysis.

# Intrusive and Non-Intrusive Instruction

# in Dynamic Skill Training *

Allen Munro
Michael R. Fehling
Douglas M. Towne

Behavioral Technology Laboratories

University of Southern California

## ABSTRACT

A distinction is drawn between computer based instruction of knowledge systems and computer based instruction of dynamic skills. There is reason to expect that the findings of research on knowledge system instruction will not apply universally to dynamic skill instruction. In particular, a theory of cognitive resource demand suggests that the principle of immediate instructional feedback may not apply in dynamic skill training. Because students in dynamic skill training are often heavily loaded with processing demands, instructional feedback must be postponed until the students have sufficient free resources to process it. This hypothesis was tested in an experiment in computer based instruction. One group of students received instructional feedback upon request, while a second group received feedback under program control. The group with control over feedback made fewer errors in training than did the group that did not control timing of the instructional feedback messages.

## Introduction

There has been little academic research on computer based dynamic skill training. Instead, the bulk of academic research on computer based learning systems has dealt with knowledge system teaching. By knowledge system we mean a coherent body of essentially propositional knowledge. By dynamic skill we mean a knowledge that coordinates perception, motor skills, and decision making in real time driven environments. Existing computer based training devices for dynamic skills are typically special-purpose vehicle simulation practice environments.

Recent advances in the development of low cost microcomputer systems make possible research on computer based dynamic skill training using systems developed soley for that research purpose. Such a low cost computer based learning research tool has been developed in our laboratory (Munro, Towne, & Fehling, 1981). This report describes the first experiment in a planned series.

It may be dangerous to assume that the research findings on computer based learning of traditional knowledge systems will apply to dynamic skill training. The present experiment was designed to explore certain contrasts between dynamic skill training and knowledge system teaching in computer based learning. Two principles of knowledge system training are challenged in the dynamic skill training environment. The first of these is that instructional feedback (knowledge of results) should be immediate. The second is that student control over instructional pacing or sequence does not ordinarily result in a large improvement in learning.

## The Dynamic Skill Training Task

The training task is loosely based upon the task of Navy Air Intercept Controllers. The task calls for close attention to the simulated radar screen, which is updated once every eight seconds. Students must quickly execute a series of key presses in response to certain events observed on the radar screen, such as the appearance of a new blip or the movement of simulated friendly and enemy aircraft into proximity. In addition, the student must keep track of certain tasks that are periodically required, such as determining fuel and weapons status for the simulated aircraft.

There are many opportunities for errors in this task. When the training system detects a student error, it can present information about the error in a reserved area of the simulated tracking computer console screen. In the first experiment, two different methods of determining when to present these messages were employed, For the intrusive feedback group, these error messages were automatically presented at the time that the system detected the errors. For the non-intrusive feedback group, an error message was shown only when requested by a student.

In a complex training session, most of a student's cognitive processing resources are likely to be allocated to attending to and responding to the task itself. If the task is suddenly intruded upon by an instructional message, the intrusion will demand additional processing resources to perform the attentional shift. This surge in processing resource demand is likely to interfere with the normal learning and performance processes. If the disruption occurs at a point in the task when a large percentage of cognitive resources are already committed, then either attention to feedback or performance on the task is likely to suffer. This attentional demand hypothesis predicts that feedback given intrusively will be more disruptive of learning than will feedback provided non-intrusively.

## The Experiment

### Method

*Subjects*. Subjects were paid volunteers who responded to posted notices and/or class announcements made at two colleges and one high school. Thirty-five subjects participated in the experiment. Of these, thirty completed the experimental training task. Subjects were assigned to one of the two experimental groups in alternating order as they arrived for the experiment. Each subject who completed the training received fifteen dollars.

*Procedure*. Subjects were run individually in the experiment. Completion of the training session required from three hours twenty minutes to four and one-half hours. All subjects first viewed a six minute videotaped explanation and demonstration of the Air Intercept Controller task. They then were instructed in the functions of each of the control devices used in the simulated task--thirteen specially labeled keyboard keys and a joystick--by a computer-based-training program called Pre-AIC. The Pre-AIC protram consisted of a series of text presentations describing the task in greater detail than had been presented in the videotaped introduction. It also presented simulation segments that the student was required to interact with by using the control keys and joystick.

After completing the Pre-AIC computer-based instruction program, all students then viewed the same videotaped sequence reviewing the Air Intercept Controller task and stating the requirements of the task. This tape segment lasted eight minutes. At this point the treatment of students in the two groups diverged. Each group viewed a videotape segment describing the way in which instructional feedback would be presented and how they should respond. Students then were given practice in the Air Intercept Controller task, using a simulator trainer program called AIC. The AIC program presented a series of 30 problems to the student, organized in four banks of five, ten, ten, and five problems. Difficulty was held roughly constant within each bank, but increased with the progression of problem banks. Students in both groups received the same problems, and the training program was the same for students in the two groups in every respect except instructional feedback.

*Instructional Feedback Treatments*. The AIC program continually monitored student performance for a variety of errors. For both feedback conditions, when the AIC program detected an error, a warning tone sounded and the word "Advisory" appeared in an area of the computer console display reserved for instructional messages. Students in the intrusive feedback group were presented with an error message for each detected new error at the time that the AIC program recognized the error. Students in the non-intrusive feedback group determined, by means of special key presses when and whether they would view the error messages.

*Data Collection*. The AIC simulation training program preserved a record of each student's interaction with the program. These data sets were later processed by data extraction programs to produce records of errors, time on problems, and other variables of interest.

*Errors*. Number of errors per problem was used as one measure of learning. Table 1 presents an analysis of variance of the error data. The mean number of errors for the studnets in the non-intrusive group was 9.17, and for the intrusive treatment group, the mean number of errors per probelm was 15.67. This difference was highly significant, suggesting that students in the non-intrusive group learned more than those in the intrusive group.

*Errors, Last Ten Problems*. A second analysis was performed, using only the error data from the last one-third of the thirty practice problems. These problems, which were longer and more difficult than most of the previous problems, were treated as a measure of final training performance in this analysis. Students in the non-intrusive feedback group made a mean of 14.74 errors per problem in these ten problems. In the intrusive group, the mean was 22.82 errors per problem.

| Intrusive Feedback Group | Non-Intrusive Group |
|---|---|
| 15.67 | 9.17 |

2-WAY ANOVA

| Source of Variation | DF | F |
|---|---|---|
| Group | 1 | 74.721** |
| Problem | 29 | 10.376 * |
| Group x Problem | 29 | 0.989ns |
| Residual | 797 | |

TABLE 1

Total Errors. Analysis of Variance

* $P<.01$
** $P<.001$

It is possible that the comparison of total errors made in practice training, given in Table 1, is not the best measure of learning. If students learn from making errors, then fewer errors in training might not portend better performance on the final task. After twenty problems, however, some training has taken place. Yet the data of Table 2 reveal that the students in the non-intrusive group made fewer errors in the last ten problems than those in the intrusive group. This result suggests that non-intrusive feedback promotes learning in dynamic skill training.

| Intrusive Feedback Group | Non-Intrusive Group |
|---|---|
| 22.82 | 14.74 |

2-WAY ANOVA
(Last 10 Problems)

| Source of Variation | DF | F |
|---|---|---|
| Group | 1 | 75.168** |
| Problem | 9 | 23.765** |
| Group x Problem | 9 | 1.245ns |
| Residual | 260 | |

TABLE 2

Errors, Last 10 Problems - Analysis of variance
** $P<.001$

Time Per Problem. The time spent on each problem by each student was recorded. Table 3 presents the analysis of the time on problem data, where time is expressed in tenths of seconds. Intrusive group students spent a mean of 264.7 seconds per problem, and non-intrusive group students took 274.3 seconds. This difference was not significant.

It had been predicted that students in the non-intrusive group would require significantly less time per problem than those in the intrusive group. In retrospect, it can be seen that this prediction is in partial conflict with the prediction of fewer errors for the non-intrusive group. Some errors, such as allowing one's fighters to be shot down by the enemy, resulted in premature termination of problem. Because intrusive group students are more likely to make such errors, a prediction of shorter time per problem for intrusive group students would have been appropriate. In any case, the experimental results do not permit the rejection of the null hypothesis that there is no difference in time to complete probelms for the two groups.

Crucial and Non-crucial Errors. Student errors are classified by the AIC program into thirty-six types. Of these, twenty-three may be termed "crucial" errors, in that they are likely to materially affect the student's chances of "winning" an exercise by shooting down the enemy aircraft. The other thirteen types of errors are non-crucial in that they reflect errors of form that will not immediately decrease the chances of winning the problem. Table 4 presents the analysis of crucial errors for all thirty problems. It shows a small but significant difference in number of crucial errors per problem between the two groups. The non-intrusive feedback group made an average 5.36 crucial errors per problem, while the intrusive group made 6.14. Table 5 shows a significant and much larger difference in non-crucial errors. The mean number of non-crucial errors per probelm for the non-intrusive training group was 3.18. For the intrusive feedback group, the mean was 9.53 errors per problem. These results suggest that, even though overloaded by the intrusive instructional messages, the intrusive group students are still able to decide which performance factors to attend to. They choose to permit greater deterioration of their non-crucial performance rather than their crucial performance.

| Intrusive Feedback Group | Non-Intrusive Group |
|---|---|
| 264.3 | 274.3 |

2-WAY ANOVA

| Source of Variation | DF | F |
|---|---|---|
| Group | 1 | 3.117 ns |
| Problem | 29 | 30.201 * |
| Group x Problem | 29 | 0.955 ns |
| Residual | 798 | |

TABLE 3

Time per Problem. Analysis of Variance.

* P<.01

| Intrusive Feedback Group | Non-Intrusive Group |
|---|---|
| 6.14 | 5.36 |

2-WAY ANOVA
FOR CRUCIAL ERRORS

| Source of Variation | DF | F |
|---|---|---|
| Group | 1 | 17.947** |
| Problem | 29 | 50.616 * |
| Group x Problem | 29 | 0.805ns |
| Residual | 797 | |

TABLE 4

Crucial Errors. Analysis of Variance.

* P<.01
** P<.001

| Intrusive Feedback Group | Non-Intrusive Group |
|---|---|
| 9.53 | 3.81 |

2-WAY ANOVA
FOR NON-CRUCIAL ERRORS

| Source of Variation | DF | F |
|---|---|---|
| Group | 1 | 67.743** |
| Problem | 29 | 3.068** |
| Group x Problem | 29 | 0.973ns |
| Residual | 797 | |

TABLE 5

Non-Crucial Errors. Analysis of Variance.

** P<.001

## Discussion

The results support our hypothesis that the processing demands of dynamic skill simulation training require non-intrusive rather than intrusive feedback. Students trained with the intrusive feedback method committed significantly more errors, both in the last set of problems and throughout the practice training, than did those trained by the non-intrusive method. Intrusive group students made significantly more errors of both crucial and non-crucial types. This higher incidence of errors is evidence that, in this dynamic skill training system, students who receive intrusive instructional feedback learn less well than those who have control over when the feedback will be presented.

Computer based training is analogous to conversation. In normal conversation, the participants take turns talking, exchanging special signals to indicate a willingness to surrender a turn or to request a turn as soon as possible. The non-intursive feedback condition simulates this aspect of conversation. The training system signals that it has a feedback message by sounding a tone. The student surrenders a turn by depressing a special key. These features allow the student to control the system's instructional input.

Our results indicate a superiority for a treatment condition that gives the student control over instructional presentation. This contrasts markedly with the results of studies of student control of conventional knowledge system instruction in CBI. For example, studies by Judd, O'Neil, & Spelt (1974); Lahey, Hurlock, & McGrain (1973); and White & Smith (1974) all find no performance improvement due to student control of instruction in computer based instruction. It is not surprising that the effect is not found in these cases, but is found in dynamic skill simulation training. In conventional computer based instruction, students consciously surrender a turn in the "conversational" interaction every time they depress the Return key.

The conversational analogy is just one way of viewing the difference between student control or lack of control of attentional demands. If the student can control the cognitive processing load by postponing instructional feedback until the processing required by the task is at a low level, then more resources will be available to process the instructional message. Furthermore, the resources devoted to the dynamic task during a demanding portion of the episode will not be diverted to processing feedback messages. Further research is called for to determine which of these effects is primarily responsible for improved learning in a non-intrusive feedback environment. It is possible that improved attention to instruction and to the task both have beneficial consequences for learning dynamic skills.

One type of error, joystick manipulation errors, required only perceptual and motor skills, rather than planning or decision making skills. Students in the two treatment groups did not differ significantly in the number of this type of error (see Table 6). This result implies that this perceptual-motor skill complex was less sensitive to the processing load imposed by the dynamic task together with instructional feedback processing than were the decision-making and planning skills called for by the task.

## Summary

The results of the intrusive feedback experiment with the AIC simulation training system do not accord with two commonly accepted precepts of computer based learning. The first of these is that instructional feedback should be as prompt as possible. In this experiment, the group that received immediate feedback performed significantly less well than the group that had the delayed feedback option. The second expectation was that student control over the instructional process would have little, if any, effect. In this experiment, the group that chose the timing of instructional feedback presentations made significantly fewer errors than the group that did not have this control. This finding is not anomalous in view of the differences in processing demand between conventional computer based instruction and dynamic skill training. The transient processing demand fluctuations of dynamic skill training can be exploited by giving students control over the timing of some aspect of instruction.

The most important conclusion to be drawn from this study is that the findings of computer based learning research on knowledge system training may not be applicable to dynamic skill training. Further research is called for to determine the characteristics of an appropriate training methodology for computer based simulation training.

## References

Judd, W.A., O'Neil, H.F., & Spelt, P.F. Individual differences and learner control I: Program development and investigation of control over Mnemonics in computer-assisted instruction. (Report No. AFHRL-TR-74-3) Lowry AFB, CO: Air Force Human Resources Laboratory, Technical Training Division.

Lahey, G.F., Hurlock, R.E., & McGrain, P. H. Post lesson remediation and student control of branching in computer based training. (Research Report SRR 73-19) San Diego, CA: Navy Personnel and Training Research Laboratory, 1973.

Munro, A., Towne, D.M., & Fehling, M.R. An experimental data collection system for research on dynamic skills training. (Technical Report No. 96) Los Angeles: University of Southern California, Behavioral Technology Laboratories, 1981.

White, A.L. & Smith, D.D. A study of the effects of varying student responsibility for instructional decisions in CAI Course. Journal of Computer-Based Instruction, 1974, 1, 13-21.

Mean Joystick Errors per Student

| Intrusive Feedback Group | Non-Intrusive Group |
|---|---|
| 31.0 | 26.6 |

T-TEST FOR JOYSTICK ERRORS

| | | | |
|---|---|---|---|
| T (pooled) | 1.03 | DF | 28 |
| T (not pooled) | 1.03 | DF | 29.96 |
| F ratio= | 0.93 | DF | 14, 14 |

Not Significant

TABLE 6

Joystick Errors. T-Test.

## Footnote

The research work described here was performed under Office of Naval Research Contract N00014-80-C-0164. Thanks are due to Henry Halff and Marshall Farr for support and advice. We thank Robert Breaux for advice and for access to training materials that helped solidify our understanding of the Air Intercept Controller task. We also thank Bob Lawson of the Office of Naval Research, Pasadena, for assistance in arranging for the observation of Air Intercept Controllers at work and at training at the Pacific Missle Test Center, Pt. Magu, and at the Fleet Combat Training Center, Pacific.

Early plans for the experimental training system used in the experiment were influenced by discussions with Michael Grady and Robin Halley of Logicon. We thank them for sharing with us some of the findings of their work on a prototype Air Intercept Controller training station.

Professors Robert Alt and Fred Shima of California State University, Dominguez

Hills, and Dean Virginia Pfiffner of El Camino College kindly granted permission for and assisted in the recruitment of student subjects from their campuses. We thank Mark C. Johnson for statistical advice and Pierre Blais for assistance in conducting the experiment.

301

# An Instructor of Associations which Learns

M. I. Chas. E. Woodson
University of California
Berkeley, CA 94720

## Abstract

The development of a general computer program which aids humans in the learning of associations is described. The program generates an instructional dialogue, carries out the dialogue, accesses the effectiveness of the instruction, and modifies its procedure for generating dialogues. Several implementations of the program improved with experience at instructing humans. In use they are now almost twice as effective as methods commonly used by college students.

### *Learning Associations*

Associations--mental connections between ideas-- are a fundamental kind of human learning. Psychologists have focused much of their study upon associations. Interestingly, there has been little study of the instructional methods for teaching associations. It would be useful to have an instructional tool which assists humans to learn associations.

In the laboratory the learning of associations is usually studied in the paired-associates task which involves two lists of items to be associated. I shall refer to the first as the *Q* list (question) and the second as the *A* list (answer) and the response made by a learner as *R* (response).

In school instruction, associations to be learned are embedded in reading and other experiences. In order to learn, the student must explicate the material to be learned from the mass of other information. Sometimes a list of pairs is given to students to memorize, but when this is done, the learners receive very little guidance as to how to do so.

Some obvious examples of associations which humans need to learn include number facts such as multiplication tables, opposites, word inflections, pairs of words from different languages, synonyms, acronyms, suffixes and meanings, prefixes and meanings, and names with phone numbers. An argument can also be made that this sort of learning underlies concept learning.

Although most instructional practice considers individual humans as the same, many psychologists and teachers believe humans are quite different in their learning behaviors and skills. The task of individualizing instruction has been an immense challenge to human instructors. With computer instructors we have the resource to seriously begin this process. With the availability of computers for instructional purposes, we can begin to treat humans as individuals. Instead of selecting a text or instructional experience for the average person or some "ideal" person, instructional experiences may be designed for individuals.

There are two extreme approaches to the design of instructional tools to aid humans learning associations.

(1) *The dumb instructor approach*. In this case an algorithm for action is programmed by an instructional programmer. A program which selects an item in sequence from a list, presents the item to the learner, receives a response from the human learner, compares that response to a key, and gives the human learner the results of the comparison, illustrates this approach. Note that no amount of experience, favorable or unfavorable, will change the behavior of the program.

(2) *The learning instructor approach*. In this case the computer program itself is capable of learning from experience. What would such an learning instructional program learn is the subject of the research reported here.

### *Instruction of Associations as a Programming Problem*

We can start with the dumb program described above which goes through the list of items presenting each question, receiving a response, and giving feedback about the response. Such a program does not learn. It is simply carrying out the operations I have programmed. It is useful, and perhaps impressive and interesting, but it is not highly intelligent.

*A System for the Use of the Program*

We can design a program to accept input of pairs from the learner. In this way, the instructional tool can be used by a learner with materials of his own choice.

We can also prepare a collection of pairs, a curriculum package. An instructional system can have a menu of sets of items available for instruction. These sets can be prepared by students or teachers. In the systems described below, we have some 50 sets of pairs available to students, with an average of about 200 items in each set, and nearly every student using the system has personal lists as well.

Some results of the use of this simple instructor are available. Generally, learning with this program (without the improvements mentioned below) by college students is about 10% more efficient than by the methods they ordinarily use. It is generally judged by students to be fun to use. The benefit to Junior High School students appears to be a bit more.

*Improvements to the simple program*

There are a number of changes we can make in the program described above to improve it.

*Random item selection.* The order in which items are entered is arbitrary and we are not interested in students learning a particular orders. Because of recency and primacy effects, it is advantageous to order items at random. Random item ordering gives a few percentage points of improvement in effectiveness.

*Repeat the difficult items more frequently.* It seems reasonable to repeat items we believe will be more difficult. Atkinson (1972) used such a system. He determined difficulty from previous experience with teaching the items to other students.

*Learner selection of Items.* Letting the learner choose the items on which he feels he needs instruction (Woodson, 1972) is another approach. This helps a bit also, particularly if learners are well-informed about strategies which lead to learning.

*Developing an intelligent instructor*

The above just about exhausts my knowledge of how to teach associations. In fact, I have more hypotheses than conclusions. This suggests I have a model but lack estimates of its parameter values. What we need is a learning instructor, which learns from experience and considers a complex array of information in making instructional decisions.

As a step to adding intelligence we can state some goals for the instructor to pursue: (1) focus upon the unlearned, (2) select steps effectively. (3) repeat effectively.

*Approaches to achieving the goals*

*Focus on unlearned.* A simple approach to this goal is to drop items from instruction on the basis of a particular number of correct responses. Instruction therefore tends to focus upon unlearned items for that person.

*Focus on unlearned.* The information in response latencies may be used to access the degree of learning.

*Orthographic prompt* may be used as an instructional step.

*Subset instruction* is another approach. Research on short term memory suggests instruction on a subset would be more effective.

*Instruction as an AI problem*

An intelligent machine may be described as having goals toward which it works. It exhibits learning when, with experience, it chooses the more effective steps to reach these goals. The goal, in this case, is to be able to effectively predict the outcomes of alternative actions. (Adding alternative actions is analogous to biological maturation in humans.) With some degree of success at predicting the outcomes of alternative instructional actions, the instructor can choose the more effective methods (or choose the less effective in an experiment to learn more about the effects of such choices).

To learn by modifying the behavior algorithm through experience requires some feedback about the adequacy of actions taken.

*Properties of an Intelligent Instructors*

1. The program has alternative behaviors (or behavior strategies) to choose among. For example, there may be a pool of instructional items from among which a presentation is to be chosen.

2. Some results regarding the effectiveness of actions chosen follows. Note this is not the case for dumb instructors. For our purposes, the improvement in the human learner's behavior is the outcome the computer instructor works toward.

3. These results regarding action choices are used to modify behavior selection process in the future.

*Need for a Theory of Learning*

To effectively choose among behavioral alternatives, we need a theory of learning. A useful theory of learning allows the instructor to effectively predict the outcomes of alternative instructional behaviors. The most favorable one can be selected.

In this case we have used a logistic model (Lord, 1968) of learning (similar to logistic psychometric models) in which items to be learned have two parameters (a difficulty and a rate of learning),

and instructional steps have two parameters (a cost and a learning outcome), and the person learning has a parameter for the degree of learning for each item.

### *Implementation of an Intelligent Instructor*

We have been able to program our instructor to do some other actions which we think will lead to learning.

Prepare and review. At night the computer instructor, reviews data from recent experience. The program tries to fit the model to better understand and predict outcomes.

Realistically, resources are limited. Time is limited. Memory is limited. Computer power is limited. Information is limited. Intelligence is limited. Therefore, an important aspect of this research is the allocation of resources. For example, what part of experience data should we record? So far, we have largely made guesses at this balance.

### *Intelligent Machines which share Knowledge*

Observation of human instructors quickly shows us that they spend time sharing what they have learned about instruction. Can we equip our intelligent machine instructor to do so?

With a family of these machines, similar in nature but functioning in different circumstances, the question arises whether these machines could share their knowledge. In some cases they have some content in common. In general, they have no people in common.

The capability of exchanging information is readily available. Off the shelf equipment is now available for computer-controlled modems, and information can be exchanged over phone lines at 1200 baud.

These computer-instructors have spare time at night and we have concentrated on machine sharing during such off hours.

It is also possible to consider the use of background processes when workload is light. We have not used this to significant degree.

### *What information to share?*

We have only begun to explore what information

may be useful to share. The following have been considered

(1) Share parameters about people in general. Much of instruction is carried out with persons for whom we have little information. Such persons are treated as "average" persons.

(2) Share parameters about items in general.

(3) Share parameters about particular item sets. When item sets are in common, the information gained may be shared.

(4) Share recollections of experience. Recorded experience might be passed among machines as paradigm cases, useful cases, or problem cases.

(5) Ask questions of each other.

(6) Send information to an archive. In this approach, one machine maintains an archive of information about instruction. The available information exceeds what can be stored and the machines must somehow choose the information to be stored.

### *AI Instructor as a Research Partner*

With a family of functioning machines we are able to ask more general questions. For example, is there an interaction between degree of previous learning and degree of prompt in an instructional exchange.

One effort we are currently exploring is research on item properties which lead to learning. We are seeking to study this question by having the instructor estimate the parameters of a learning model for sets of items with each property.

### *Conclusions*

We have described the development of an intelligent learning device, which makes complex decisions while carrying out instruction, which benefits from experience, and which shares its learning with others of similar nature. The devices are learning from experience and results show they are about 80% more effective than the usual methods of learning associations. Ways to have the programs share their learning are being explored. Might they learn and then teach humans what they have learned about learning?

### References

Anderson, J. H. & Bower, G. H. *Human Associative Memory*. New York: John Wiley, 1973.

Atkinson, R. C. Optimizing the learning of a second-language vocabulary. *Journal of Experimental Psychology*, 1972, *96*, 124-129.

Lord, F. M. & Novick, M. R. *Statistical theories of mental test scores*. Reading, Mass: Addison-Wesley, 1968.

Woodson, M. I. C. E. Learner judgment in instructional decisions for learning meaningful paired associates. *Journal of Experimental Psychology*, 1974, *102*, 167-169.

# A TRIP TO EUROPE ... APPAREL TERMINOLOGY IN SIX FOREIGN LANGUAGES

KAREN E. SCHAEFFER

University of Delaware

ABSTRACT - A Trip to Europe ... Apparel Terminology in Six Foreign Languages

The PLATO program "Trip to Europe" was developed in the Department of Textiles, Design and Consumer Economics at the University of Delaware to assist students who elect the Winterim course offering "European Textiles and Fashion Study Tour." As part of the total requirement, students must complete a research project on a clothing or textile topic of their choice. Because people within the major are not required to have a foreign language, this program, which includes twenty-four apparel terms in six Western European languages was written. The primary objective of the program was to expedite information gathering by students for their clothing or textiles project.

## INTRODUCTION

Biennially, during the January Winterim Session, the Textiles, Design and Consumer Economics Department in the College of Human Resources at the University of Delaware offers the European Textiles and Fashion Study Tour and an accompanying Seminar.

Students in the Department and especially those having their major emphasis in design, fashion merchandising, or textile technology can elect to take the Tour for a total of six credits. Professional appointments fall, roughly, into the following categories - costume design, apparel/or apparel related manufacturing and merchandising, and historic costume.

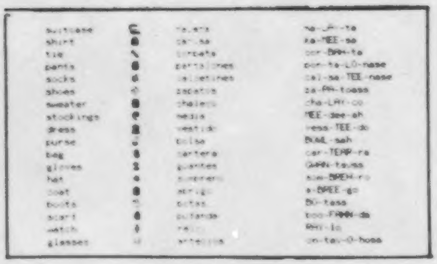
While on the trip students are required to complete an in depth paper on a clothing or textile topic of their choice. In preparation, during the preceding Fall, students submit a prospectus of their topic to the faculty member responsible. In depth research of the topic, however, is completed during the three and one half weeks spent in Europe.

## PURPOSE

To assist these students who do not have a foreign language requirement, a PLATO lesson was designed entitled "A Trip to Europe." The purpose of the lesson is to give pupils the ability to recognize and use apparel terminology in six Western European languages - Italian, Spanish, French, Dutch, Swedish, and German. Familiarization with the 18 apparel terms in the languages of countries on the tour will expedite information gathering for the projects.

## INDEX

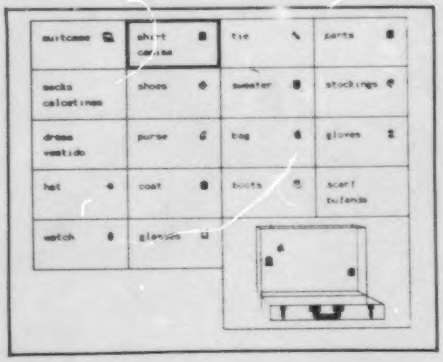
The students begin the lesson by selecting one of the six language options. Following this selection, an index page of the foreign language vocabulary selected appears, having on the left side of the screen, the English word and a small diagram of the apparel item. The center of the screen has the same word in the selected foreign language, and the right side of the screen has the syllabalized, phonetic pronunciation of the word. (See Figure I) As the pupils progress through the lesson, they can always return to this page via the HELP key. The lesson itself is broken into three segments and is completed in a story format.



| | | |
|---|---|---|
| suitcase | maleta | ma-LAY-ta |
| shirt | camisa | ka-MEE-sa |
| tie | corbata | cor-BAH-ta |
| pants | pantalones | pon-ta-LO-nase |
| socks | calcetines | cal-se-TEE-nase |
| shoes | zapatos | za-PA-toass |
| sweater | chaleco | cha-LAY-co |
| stockings | media | MEE-dee-ah |
| dress | vestido | vess-TEE-do |
| purse | bolsa | BOAL-sah |
| bag | cartera | car-TERR-ra |
| gloves | guantes | GWAN-tauss |
| hat | sombrero | som-BREH-ro |
| coat | abrigo | a-BREE-go |
| boots | botas | BO-tass |
| scarf | bufanda | boo-FAHN-da |
| watch | reloj | RAY-lo |
| glasses | anteojos | an-tay-O-hoss |

FIGURE I

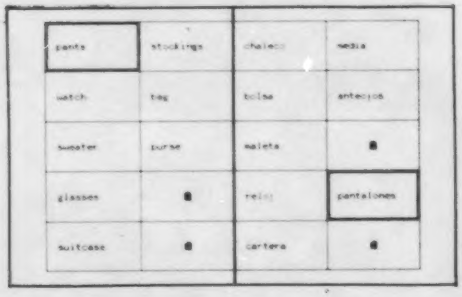
Part I introduces the students to the scenerio. The screen is segmented into 18 rectangles where in each appears the English apparel word accompanied by a small drawing. In the lower right hand side of the screen a suitcase appears. (See Figure II) Pupils are instructed to "pack the suitcase" by touching the rectangles and then the suitcase. The small drawing then appears in the suitcase while the foreign term appears in the rectangle below the English word. This procedure is repeated until all 18 apparel item drawings are in the suitcase.



| | | | |
|---|---|---|---|
| suitcase | shirt<br>camisa | tie | pants |
| socks<br>calcetines | shoes | sweater | stockings |
| dress<br>vestido | purse | bag | gloves |
| hat | coat | boots | scarf<br>bufanda |
| watch | glasses | | |

FIGURE II

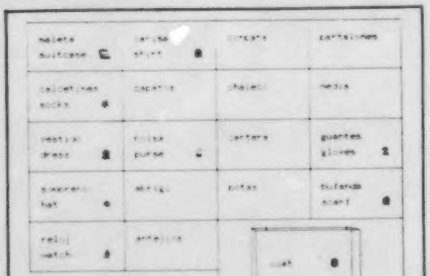
Part II consists of a drill giving students the opportunity to work with the new foreign term and its English equivalent. Ten apparel terms in English appear on the left side of the screen in rectangles similar to those in Part I. The foreign word equivalents appear randomly on the right side of the screen in a similar manner. (See Figure III) Students then have to touch the two words that they think correspond and then the NEXT key. If the match is accurately made, the apparel symbols appear in both boxes simultaneously. If the match is not made correctly, the words will remain in the rectangles. This exercise can be repeated as many times as necessary by pressing the DATA key; a new word mix is presented each time the key is pressed.



| | | | |
|---|---|---|---|
| pants | stockings | chaleco | media |
| watch | bag | bolsa | anteojos |
| sweater | purse | maleta | |
| glasses | | reloj | pantalones |
| suitcase | | cartera | |

FIGURE III

The storyline of Part III has the pupils arriving in the language appropriate foreign country. In essence, the student has to unpack the suitcase using the foreign terms. Now, the 18 foreign apparel terms appear in the rectangles on the screen. (See Figure IV) The suitcase, again in the lower right hand side of the screen, has the apparel drawing. Students are instructed to touch the rectangle containing the appropriate foreign term. When this is completed correctly, the English term will appear underneath the foreign word in the rectangle, and a new apparel drawing will appear in the suitcase. If pupils do not touch the correct box containing the corresponding foreign term, the foreign term will remain until the match is appropriately made. As in Part II, this portion of the lesson can be repeated by pressing the DATA key. As indicated earlier, the students can, at any time during the lesson, press HELP and go back to the vocabulary list presented at the outset of the program, in the event that frustration sets in. The final frame rewards the students by congratulating them in the foreign language and indicating that they have successfully completed the lesson.



| | | | |
|---|---|---|---|
| maleta<br>suitcase | camisa<br>shirt | corbata | pantalones |
| calcetines<br>socks | zapatos | chaleco | media |
| vestido<br>dress | bolsa<br>purse | cartera | guantes<br>gloves |
| sombrero<br>hat | abrigo | botas | bufanda<br>scarf |
| reloj<br>watch | anteojos | coat | |

FIGURE IV

The reaction of the initial group of students to the lesson has been favorable.

Their feedback indicated that PLATO was an effective teaching mechanism for the terminology. Although the structure of the lesson was simplistic, students felt that learning language terms in this way was beneficial. Additional suggestions from them included expansion of the lesson using everyday expressions, short phrases and food terminology.

The author currently plans to increase the number of apparel terms used in the program for the group of students who will be taking the trip next time. Some of the student suggestions may be included as well.

# A TRIP TO EUROPE ... APPAREL TERMINOLOGY IN SIX FOREIGN LANGUAGES

KAREN E. SCHAEFFER

University of Delaware

ABSTRACT - A Trip to Europe ... Apparel Terminology in Six Foreign Languages

The PLATO program "Trip to Europe" was developed in the Department of Textiles, Design and Consumer Economics at the University of Delaware to assist students who elect the Winterim course offering "European Textiles and Fashion Study Tour." As part of the total requirement, students must complete a research project on a clothing or textile topic of their choice. Because people within the major are not required to have a foreign language, this program, which includes twenty-four apparel terms in six Western European languages was written. The primary objective of the program was to expedite information gathering by students for their clothing or textiles project.

## INTRODUCTION

Biennially, during the January Winterim Session, the Textiles, Design and Consumer Economics Department in the College of Human Resources at the University of Delaware offers the European Textiles and Fashion Study Tour and an accompanying Seminar.

Students in the Department and especially those having their major emphasis in design, fashion merchandising, or textile technology can elect to take the Tour for a total of six credits. Professional appointments fall, roughly, into the following categories - costume design, apparel/or apparel related manufacturing and merchandising, and historic costume.

While on the trip students are required to complete an in depth paper on a clothing or textile topic of their choice. In preparation, during the preceding Fall, students submit a prospectus of their topic to the faculty member responsible. In depth research of the topic, however, is completed during the three and one half weeks spent in Europe.

## PURPOSE

To assist these students who do not have a foreign language requirement, a PLATO lesson was designed entitled "A Trip to Europe." The purpose of the lesson is to give pupils the ability to recognize and use apparel terminology in six Western European languages - Italian, Spanish, French, Dutch, Swedish, and German. Familiarization with the 18 apparel terms in the languages of countries on the tour will expedite information gathering for the projects.

## INDEX

The students begin the lesson by selecting one of the six language options. Following this selection, an index page of the foreign language vocabulary selected appears, having on the left side of the screen, the English word and a small diagram of the apparel item. The center of the screen has the same word in the selected foreign language, and the right side of the screen has the syllabalized,phonetic pronunciation of the word. (See Figure I) As the pupils progress through the lesson, they can always return to this page via the HELP key. The lesson itself is broken into three segments and is completed in a story format.

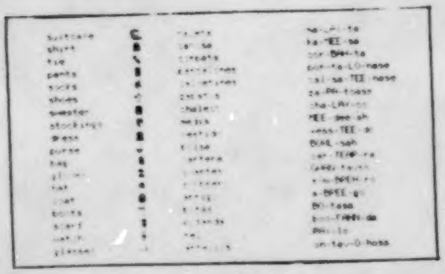


FIGURE I

Part I introduces the students to the scenerio. The screen is segmented into 18 rectangles where in each appears the English apparel word accompanied by a small drawing. In the lower right hand side of the screen a suitcase appears. (See Figure II) Pupils are instructed to "pack the suitcase" by touching the rectangles and then the suitcase. The small drawing then appears in the suitcase while the foreign term appears in the rectangle below the English word. This procedure is repeated until all 18 apparel item drawings are in the suitcase.

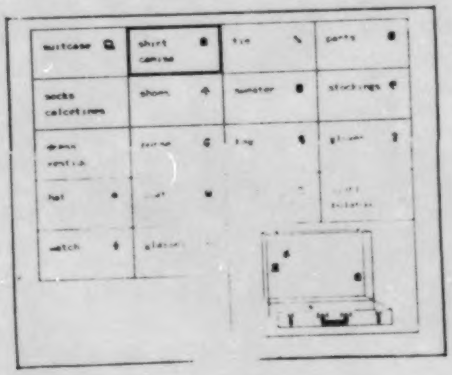


FIGURE II

Part II consists of a drill giving students the opportunity to work with the new foreign term and its English equivalent. Ten apparel terms in English appear on the left side of the screen in rectangles similar to those in Part I. The foreign word equivalents appear randomly on the right side of the screen in a similar manner. (See Figure III) Students then have to touch the two words that they think correspond and then the NEXT key. If the match is accurately made, the apparel symbols appear in both boxes simultaneously. If the match is not made correctly, the words will remain in the rectangles. This exercise can be repeated as many times as necessary by pressing the DATA key; a new word mix is presented each time the key is pressed.

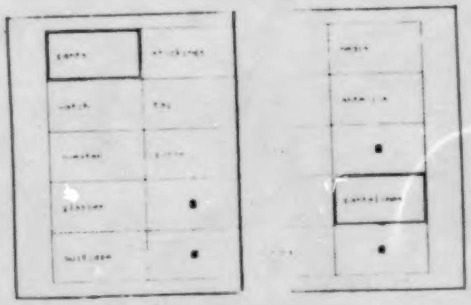


FIGURE III

The storyline of Part III has the pupils arriving in the language appropriate foreign country. In essence, the student has to unpack the suitcase using the foreign terms. Now, the 18 foreign apparel terms appear in the rectangles on the screen. (See Figure IV) The suitcase, again in the lower right hand side of the screen, has the apparel drawing. Students are instructed to touch the rectangle containing the appropriate foreign term. When this is completed correctly, the English term will appear underneath the foreign word in the rectangle, and a new apparel drawing will appear in the suitcase. If pupils do not touch the correct box containing the corresponding foreign term, the foreign term will remain until the match is appropriately made. As in Part II, this portion of the lesson can be repeated by pressing the DATA key. As indicated earlier, the students can, at any time during the lesson, press HELP and go back to the vocabulary list presented at the outset of the program, in the event that frustration sets in. The final frame rewards the students by congratulating them in the foreign language and indicating that they have successfully completed the lesson.

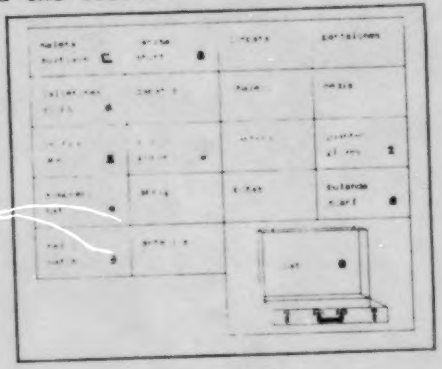


FIGURE IV

The reaction of the initial group of students to the lesson has been favorable.

Their feedback indicated that PLATO was an effective teaching mechanism for the terminology. Although the structure of the lesson was simplistic, students felt that learning language terms in this way was beneficial. Additional suggestions from them included expansion of the lesson using everyday expressions, short phrases, and food terminology.

The author currently plans to increase the number of apparel terms used in the program for the group of students who will be taking the trip next time. Some of the student suggestions may be included as well.

# COMPUTER-BASED INSTRUCTION OF EXPERT PROBLEM SOLVING PROCEDURES: ENSURING ACCESS TO RELEVANT KNOWLEDGE STRUCTURES

Richard W. Evans, Ph.D. - The Learning Center

State University of New York College at Farmingdale

One of the primary goals of a higher education is the ability to solve problems (Simon, 1980). This is particularly the case in a modern, highly technologized society such as our own where the demands of the work and social environments are continually changing. Interest in the teaching of problem solving abilities has lately been sharply on the rise, and the development of successful techniques for the classroom teaching of problem solving ability has been reported (Hayes, 1980). In addition, techniques of problem solving have been taught in a computer-based form (Goldstein, 1980). The application of the computer to the teaching of problem solving ability is especially intriguing. Our understanding of problem solving and the techniques for teaching it is still in its early stages, and considerable research remains to be done in this area. Since intensive research is costly, such a program requires a medium such as the computer as its potential for inexpensive and widespread dissemination of successful teaching methodology makes the enterprise economically feasible (Rutherford, 1981).

While much of the work in problem solving has concentrated on the creative aspects of problem solving, the focus of this paper is primarily on the procedures employed by expert problem solvers within particular knowledge domains, such as mathematics, chess, or physics. Cognitive psychologists have recently succeeded in providing quite specific descriptions of the forms of knowledge and problem solving procedures employed by experts in such knowledge domains. This is particularly true of physics (Larkin, McDermott, Simon and Simon, 1980; Chi, Feltovich and Glaser, 1980). The identification and description of these specific problem solving abilities and the development of the sophisticated methods needed to discover them suggests that this approach can be extended to a number of domains where the problems encountered tend to be well-defined. If such expert knowledge representations and the problem solving procedures which they produce can be described for a large number of domains, then there looms the possibility that a complete revolution in the teaching of domain-specific problem solving abilities is in the making. In order to teach such problem solving abilities, a second enterprise is called for: the techniques which are most effective in imparting expert knowledge must be determined.

This enterprise has been labelled "Cognitive Instructional Science" (Resnick, 1980). Once a sufficient description of the expert knowledge structures, which are employed in problem solving, has been made, then it is the task of Cognitive Instructional Science to discover those techniques which best impart these knowledge structures to the learner. This is essentially a two-fold task. On the one hand, it is necessary to find ways of helping the learner build the appropriate knowledge structures. This may involve discovering the appropriate mode of presentation, analyzing the learner' characteristics, and examining other cognitive learning factors. The second aspect of this enterprise is the discovery of those techniques which ensure that the knowledge structure created is accessible (that is, can be retrieved from memory) when it is needed in solving a problem. It is to this problem, the problem of ensuring accessible expert knowledge structures, that the presen paper is addressed.

Accessing Memory: A great deal of research has been conducted on the structure and functioning of human memory over the past twenty years. While much of this research has concentrated on determining precise stages of memory processing, there has emerged from the variety of viewpoints and theories at least one very prominent principle of memory: human memory is highly sensitive to context. An item which is appropriately embedded in meaningful material will be stored in memory and recalled later when the overall context is re-created. It is this reconstruction of the original context which is essential to the point of this paper. In order to assure that a given aspect of a knowledge structure be accessed when needed, it must be stored in a context which will be present at the time it is needed for recall. This meaningful context will access the relevant memory structure. There are innumerable examples of this principle of memory access in the psychological literature. Mnemonic devices generally involve providing such a context for items to be recalled; the subsequent regeneration of the device (for example a set of loci) accesses the stored items in a manner which is often surprisingly easy (Bower, 1970). These meaningful contexts can take on a variety of forms from single words (Tulving and Pearlstone, 1966) to entire verbal passages (Bransford and Johnson, 1973), to visual scenes (Shepard, 1967) and even environmental situations (Baddeley, 1976). In order to build expert knowledge structures which are accessible at appropriate times during problem solving, it is necessary to design instructional materials which will build and store the knowledge structure representing the particular problem solving procedure within the context of the problem or sub-problem to which that problem solving procedure is relevant. This approach to the generation of accessible expert knowledge structures is here referred to as "Problem-Oriented Instruction".

Problem-Oriented Instruction: In Problem-Oriented Instruction, the various aspects of the knowledge structure to be acquired are each taught within the context of the problems or sub-problems to which they apply. As a result, following the principle of human memory operation described above, an occurance of the problem or sub-problem at a later time will constitute a contextual situation which accesses the appropriate aspect of the knowledge structure. When accessed in this way, the knowledge structure will generate those problem solving procedures relevant to the problem at hand. This general approach, when modelled on an expert's problem solving procedures, should be capable of storing expert knowledge structures in such a way that they are accessed under the same situations that an expert would access them. The result should be a student who very quickly acquires expert problem solving ability.

In the study to be described, some simplifying assumptions about expert problem solving have been made, especially as applies to the particular knowledge domain of concern - elementary statistics. To begin with it is assumed that the expert problem solver approaches similar problems in similar ways - making certain decisions first, and generally following a succession of problem-solving procedures which tend to be repeated in similar situations. This repeating order of procedures is suggested by the studies performed with physics problem solving experts and reflects the underlying assumption that any problem solving expert owes much of his expertise to the fact that he or she seldom enters blind problem solving "alleys" requiring a regression. Thus the expert problem solver makes "basic" decisions about his approach to a particular problem (Chi, et.al., 1980) such that for a particular type of situation, the succession of problem solving procedures employed is such as to make continual progress in approaching a solution. This assumption has implications for the way in which access to a knowledge structure is best developed. By assuming that the expert has an efficient succession of access to relevant knowledge structures for each type of problem situation, the problem-oriented instructional units employed in this study not only store each solution procedure in the context of the problem to which it is relevant, but also store each succeeding procedure in the context of the preceding procedure. Thus there is a sequence of memory access such that a problem accesses a sequence in the knowledge structure which causes a search of memory until the problem type is recognized; this search and subsequent successful memory match (recognition) then accesses the next appropriate solution procedure. This procedure operates on the problem, transforming it so that it becomes the context which activates the next relevant solution procedure and so on, until a final solution to the problem is reached. The following study describes the employment of this general approach to Problem-Oriented Instruction applied to a computer-based course in elementary statistics.

## Method

Materials and Equipment: Sixteen instructional units of about 2 hours duration each were prepared for presentation on a DEC PDP 11-45 computer equipped with 12 VT-100 video terminals. The units were designed to present a full course in elementary statistics, following essentially the same structure and organization commonly found in textbooks, covering the range of topics from simple elementary descriptive statistics to one-way analysis of variance. The CAI language employed was one developed specifically for the project and was very similar to PILOT. Use of graphics was common and employed DEC's basic graphics package. The approach taken in designing the individual units was one which gradually evolved, and the process is described more fully elsewhere (Evans, 1981). Most of the units contained significant segments which followed the problem solving procedures of an expert in the field (the author) as determined by a protocol analysis procedure (Newell and Simon, 1972). As such, large portions of the computer-based units were designed following Problem-Oriented Instruction guidelines and would be expected to produce knowledge structures which would be readily accessed by relevant problems.

In following this design procedure, the units typically began by introducing students to a problem, the solution of which was the topic of the instructional unit. The student was then guided, through an interactive process, in working through a complete solution to the problem. The units consisted primarily (over 60%) of this initial problem solving procedure, with the remainder of the unit concerned with extensions of this basic procedure. Following is an example demonstrating this type of instructional procedure for the topic of variability. In this case the problem is that of numerically representing variability. The example below assumes a very cooperative student.

- - - -

(Computer) Suppose for the moment that you are engaged to conduct a study examining the short-term effects of alcohol on memory. In one group (the Experimental Group), subjects drink an ounce of alcohol 30 minutes prior to engaging in your memory task. The second group of subjects (the Control Group), drinks an equal amount of water before taking the memory test. The memory test consists of a single 15-item list of words. The results of the task for the two groups follows:

| | Number of words recalled | |
|---|---|---|
| Subject | Expt. Group | Cont. Group |
| # 1 | 10 | 14 |
| 2 | 8 | 6 |
| 3 | 10 | 13 |
| 4 | 9 | 7 |
| 5 | 11 | 8 |
| 6 | 12 | 12 |
| 7 | 11 | 10 |
| 8 | 9 | 8 |
| 9 | 10 | 12 |

Now study these results for a minute.
(hit 'carriage return' to continue)

(Student) Hits Carriage Return.

(Computer) OK, now tell me - as the experimenter, would you conclude from these results that alcohol has no effect on memory?

(Student) "No" Hits Carriage Return.

(Computer) Oh? Well, let's see. Tell me this - do the means of the two groups differ from each other?

(Student) "No." Hits Carriage Return.

(Computer) Very good. You're right about that. So look, we have just found that the mean for these two groups is the same - doesn't that mean that alcohol had no effect on memory in your experiment?

(Student) "No". Hits Carriage Return.

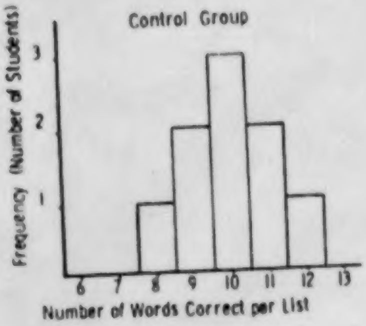
(Computer) OK, so you think there is something in the results that shows a difference between the two groups which is not reflected in the mean?

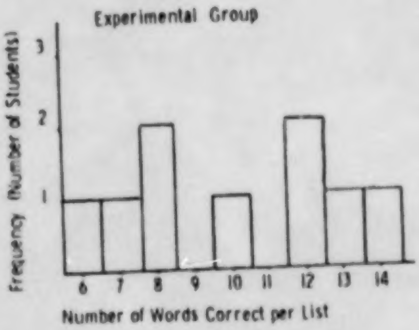
(Student) "Yes". Hits Carriage Return.

(Computer) Very good. I think I agree with you. Shall we graph these results to get a slightly different look at them?

(Student) "Yes". Hits Carriage Return.

(Computer) OK, here they are.





Tell me, do these graphs look correct to you?(computer is displaying both graphs and tabular data simultaneously).
(Student) "Yes" Hits Carriage Return.
(Computer) You're right. I've put no mistakes in these graphs. Let's examine them then. We know that the mean, or middle point of each distribution is the same. But you suggested that there is some other difference between the two groups. Do you see that difference in these graphs?
(Student) "Yes". Hits Carriage Return.
(Computer) I agree. There IS a difference between the two groups which the mean does not capture. In the case here, the subjects in group two are more different from each other than the subjects in group one (the control group) In other words they vary more from each other. Does it seem likely then, that the alcohol has affected memory by making it more variable?
(Student) "Yes". Hits Carriage Return.
(Computer) OK. Very good. Now we need to figure out how to quantify these differences in variability.

- - - -

The above example is much simplified, but it does serve to illustrate the main aspects of Problem-Oriented Instruction. The simplifications generally involve assumptions about the student's rate of progress. For example, a large number of students will fail to see that the two groups are different even though the means are the same. There would have to be another segment of the instructional unit to which the program would branch in the event that the student answered "No" to the question regarding the effect of alcohol given that the means were the same. Despite the simplifications involved, the example does illustrate the style of instruction employed here, and it accomplishes several things. First, the student begins to learn the material within the context of a problem: there are two groups of subjects which appear to differ from each other, but their average performance is the same. Within this context, the student learns some simple problem-solving procedures designed to ascertain the nature of the difference between the groups - he can view graphs of their results. As a result the student is building a knowledge structure within the context of the main problem and additional subproblems, such that each principle for problem solution has been stored within the context to which it is relevant. Thus whenever comparing two groups in the future, the student will be dealing with a context which is likely to access, among other procedures, the procedure which calls him to check for variability. Performing such a check is likely to access again the procedure for constructing and examining graphical representations of these results. The remainder of the unit will follow a similar approach, with the result that the student will have stored, within the overall context of comparing different groups of subjects and the succeeding context of finding differences in variability, procedures for deciding what numerical measure of variability is appropriate, and within that context, procedures for calculating the appropriate variability measure. As a result, it may be that the student will perform efficiently and solve problems in a manner quite similar to an expert in the field.

Subjects: Subjects in the study were sophomore and junior psychology majors and minors enrolled in the Psychological Statistics course in the Psychology Department at Beaver College, Glenside, Pennsylvania. Approximately 36 students began the computer-based course. Another 20 students from the previous year's course participated in the evaluation.

Evaluation: Several different forms of evaluation were employed in assessing the effects of the computer-based course. As the computer was the only interactive form of instruction which the students had, there was no effect possible from lectures, though students had some contact with peer tutors when tested on the individual units. In general, then, any performance changes from previous years are likely to be due to the computer-based lessons. Students were administered rating scales for evaluation purposes at the conclusion of each unit mastery test and also at the conclusion of each semester. In addition, a check was made of retention rates (number of students enrolled over the duration of the course) and also rate of progress as measured by the calendar dates on which the corresponding unit tests were taken and passed. A final evaluation measure was a statistical problem solving test, designed by the author, to assess problem solving performance on ten different areas within elementary statistics. This test was presented in multiple-choice format both to students who had taken the computer-based course and to students who had previously taken the statistics course in Keller-method format.

## Results

Student Evaluation: Analysis of the questionnaires revealed that the students significantly preferred the computer as a source of information to the workbooks available for study in the Keller method program. In addition, students found that the computer was as valuable in providing information as were the workbooks.

Retention Rates and Rate of Progress: By comparing the rates at which students dropped out of the computer-based course and out of the Keller method course (for the period covering the two years prior to the computer-based course), Figure 2 shows that there is essentially no difference in student retention between the two courses. When the regression line slopes are calculated and adjusted to give the proportional rates of loss per unit, this figure is .029 for the computer-based course, and .028 for the Keller method course. Examining the speed at which students progressed through the course, as revealed by the average day of year on which individual unit exams were taken, it is clear that there is no significant difference between rates of progress in the computer-based and Keller-method courses, as shown in Figure 3. The mean test completion date across all units for the computer-based course is day 208.9 while for the Keller method course it is 211.1.

Statistical Problem Solving Ability: The statistical Problem Solving Test was administered to students in the computer-based course and to students from the previous Keller method course. Results on this 10-item test gave an average of 6.61 items correct for students taking the computer-based course, and an average of 4.64 items correct for students taking the Keller method course. A Student's t-test performed on the data yielded a t of 9.80 (df = 46, $p < .001$) indicating a highly significant improvement on problem solving ability for students in the computer-based course.

## Discussion

The computer-based course had a highly significant effect in improving students' statistical problem solving ability. Since there was no effect on retention rates and rate of progress, it is unlikely that the difference between the computer-based and the Keller-method course is due to motivational factors. Rather, the improvement in problem solving ability reflects the fact that the instructional units in the computer-based course predominantly are designed according to Problem-Oriented Instruction, and thus are providing students with problem-solving procedures stored within the context of the problems to which they are relevant. Thus the student does not possess a sterile form of knowledge: instead the student finds that when a problem is encountered, he can readily access a problem solving procedure for efficiently dealing with that problem. It is reasonable to conclude then, that by designing computer-based materials in this Problem-Oriented manner, instructional materials can be created which provide a student with a knowledge structure which, like an expert's, will be accessed efficiently and at appropriate times.

It was suggested earlier that there are two aspects of Cognitive Instructional Science. The first is the training of expert knowledge structures - finding ways of designing and presenting information so that students can readily develop representations of it. The second aspect of Cognitive Instructional Science has the task of ensuring that these representations are usefully employed by being accessed at appropriate times - employing the rules of human memory access to order and coordinate the individual instructional segments in a complete, cognitively-designed program. The approach taken to the second of these aspects of Cognitive Instructional Science is to teach these knowledge structures within the context of the situations in which they will be employed. The study reported has demonstrated the efficacy of this approach. A major weakness, however, is the lack of sufficient research into the expert knowledge structures employed in statistical problem solving. In the study reported, the form of these representations was generally not known. As a result, important information for the implementation of cognitive instructional design in a computer-based course on statistical problem solving was missing. If this information had been available, presumably the results, while highly significant, would have been more impressive still. Further research remains to be done before the goal of designing a fully effective computer-based course in elementary statistics is achieved. Nevertheless, as the present study demonstrates, the general approach to training expert problem solving is effective, and requires only time and further research to be fully realized.

## References

Baddeley, A.D., 1976, The Psychology of Memory. New York: Basic Books.

Bower, G.H., 1970, Analysis of a mnemonic device. American Scientist, 58,495-510.

Bransford, J.D., and Johnson,M.K., Considerations of some problems of comprehension. In W.G. Chase (ed.), Visual Information Processing. New York,Academic Press, 1973.

Chi,M.T.H., Feltovich,P., Glaser,R. 1980. Representation of Physics Knowledge by Experts and Novices. ONR Technical Report No. 2, Learning Research and Development Center, University of Pittsburgh.

Evans, R.W., 1981, A Computerized course in elementary statistics: Educational objectives and methods. In Proceedings of NECC 1981: National Educational Computing Conference; p. 254 - 258.

Goldstein, I., 1980. Developing a computational representation for problem-solving skills. In Tuma,P. and Reif,F. (eds), Problem Solving and Education: Issues in Teaching and Research. Hillsdale, N.J. Lawrence Erlbaum Associates.

Hayes, J.R., 1980, Teaching problem-solving mechanisms. In Tuma,D. and Reif,F. (eds) Problem Solving and Education: Issues in Teaching and Research. Hillsdale, N.J. Lawrence Erlbaum Associates.

Larkin, J.H., McDermott, J., Simon, D.P., and Simon, H.A., 1980, Expert and Novice performance in solving physics problems. Science, 80, (4450):1335-42.

Newell, A., Simon, H.A., 1972. Human Problem Solving. Englewood Cliffs, N.J. Prentice Hall.

Resnick, L., 1980, Instructional Psychology Annual Review of Psychology, July.

Rutherford, F.J., Computing and Technology for the 80's. Keynote address, delivered at the National Educational Computing Conference, Denton, Texas, June,17 - 19, 1981.

Shepard, R.N., 1967, Recognition memory for words, sentences, and pictures, Journal of Verbal Learning and Verbal Behavior, 6:156-163.

Simon, H.A., 1980, Problem Solving and Education. In Tuma,D., and Reif,F. (eds) Problem Solving and Education: Issues in Teaching and Research. Hillsdale, N.J. Lawrence Erlbaum Associates.

Tulving, E., and Pearlstone, A., 1966, Availability vs. accessibility of information available for words. Journal of Verbal Learning and Verbal Behavior. 5: 381-91.

## Notes

1. The work reported in this paper was supported by a National Science Foundation LOCI (Local Course Improvement) grant to Dr. Bernard M. Mausner, Psychology Department, Beaver College, Glenside Pa. The author served as Principal Investigator on that grant. In addition, Dr. Edward N. Wolff, of the Mathematics Department, Beaver College, served as systems programmer.

2. The author would like to acknowledge the assistance of Mary M. DeBoer in the data collection and analysis reported in this study. He is further indebted to Ms. DeBoer and to Anita L. D'Amore for their assistance in programming and editing the instructional units employed.

Figure 2
Retention Rates for the Computer-Based and the Keller Method Statistics Courses

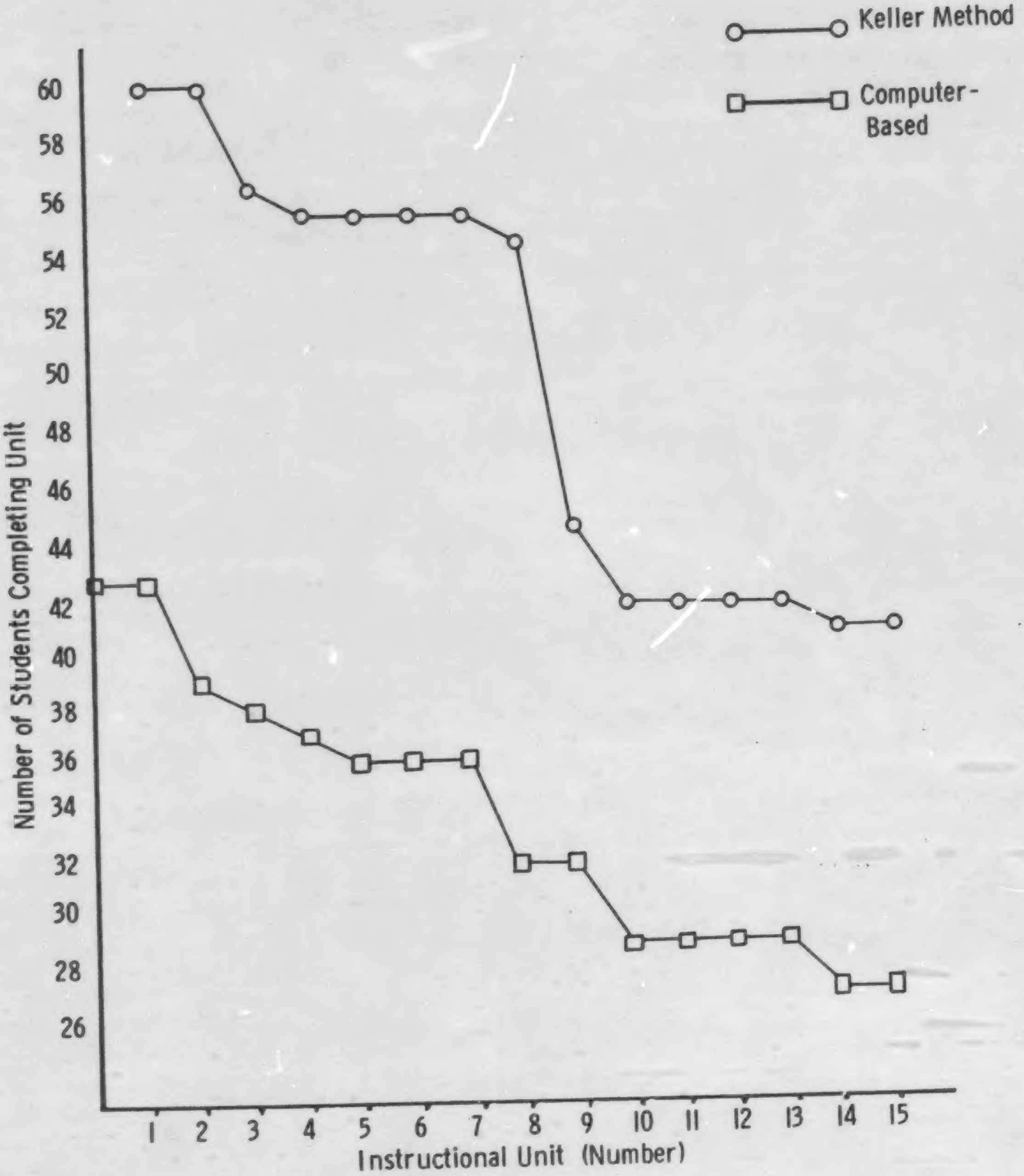
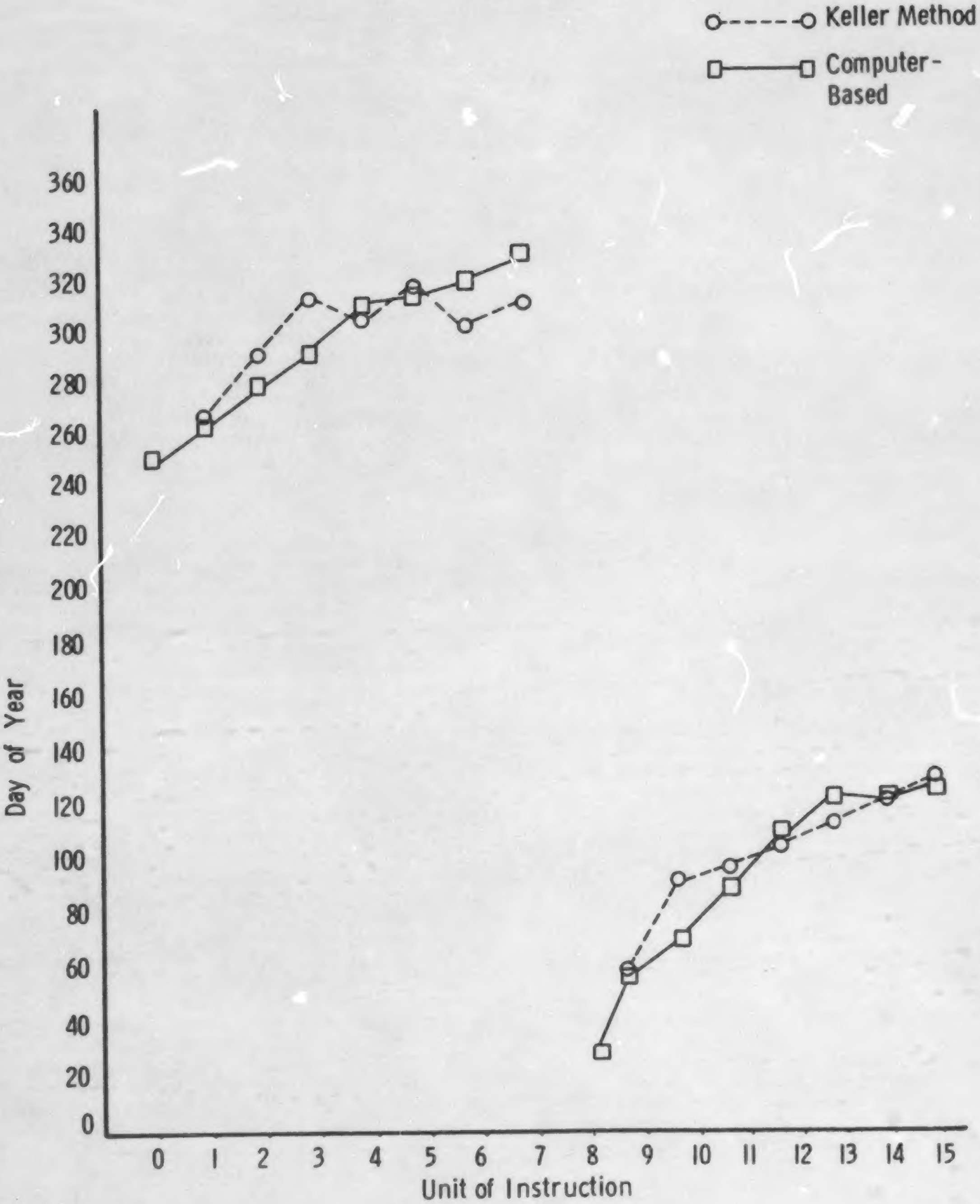


Figure 3
Progress Rates for the Computer-Based and the Keller Method Statistics Courses



# FORCED CORRECTION IN COORDINATE CONCEPT LEARNING

Timothy W. Spannaus, Ph.D.

Creative Universal, Inc.

Many of us who have been through several years of computer-assisted instruction have run up against the frustration of the common requirement to enter a correct answer before proceeding to the next screen.

While most authors of CAI design texts recommend providing some sort of exit, to avoid frustration, the question of the effectiveness of requiring that correct answer remains (Sherwood, 1974; Schaefges, Eisenberg & Stock, 1980).

## BACKGROUND OF THE PROBLEM

The roots of the Forced Correction tactic can be traced from programmed instruction to computer-assisted instruction.

### The Influence of Programmed Instruction

In the classic Skinnerian program, the requirement of correct response fell on the designer of the programmed text. It was suggested that, in a well-designed program, every student would respond correctly at every frame (Skinner, 1966). There were two bases for this requirement:

1. Each frame was preceded by all the necessary prerequisites, so there is no reason anyone should err;
2. Each frame, in turn, provided prerequisites for succeeding frames, so mastery of each frame was essential to the frames that follow.

Given the difficulty of writing error-free programmed instruction, the requirement became, in CAI, a function of the authoring language. Many languages, including Tutor, hold a student at a response point until the correct response is provided.

The purpose of the present study, then, is to examine the usefulness of the Forced Correction procedure. To be certain that the treatment does not itself introduce confounding problems, a well-validated instructional design model was used, based on the work of M. David Merrill in general and the coordinate concept learning model in particular (Merrill & Tennyson, 1977).

### The Instructional Model

The model defines coordinate concepts in terms of other concepts. That is, if one can break down a concept into a set of subordinate concepts, the subordinate concepts are coordinate. In a classic example, trochaic and iambic are coordinate concepts, subordinate to the concept of poetic foot. Concepts are defined as a set of specific objects, symbols or events which share common characteristics and which can be referenced by a particular name or symbol (Tennyson & Park, 1980).

The basic model is an expansion of the so-called Ruleg model from programmed instruction. After many years of debate and research, the programmed instruction fraternity determined that, in concept learning, there is a decided advantage to presenting the rule, or generalization, first. followed by examples (Markle, 1969, 1978). Since that time, the exact composition of the example set has been determined (Tennyson & Park, 1980). Merrill and his colleagues have differentiated examples and practice items, so the present model emerges as Rule, Example, Practice (Merrill & Tennyson, 1977).

## METHODOLOGY

The present study uses a completely randomized design, with two treatment groups. Both groups studied a specially-constructed CAI lesson on the coordinate concepts of measures of test reliability.

While participants were randomly assigned to treatment groups, all students were administered a selection test. One portion of the selection test was designed to measure attainment of a prerequisite skill, interpretation of correlation coefficients. Those who failed this portion of the test were routed to a brief lesson on that skill. A second portion of the selection test measured mastery of the concepts to be taught, since there would be little point in attempting to differentiate student performance if the sample included students who had mastered the concepts.

The lesson was constructed following a mastery model, so that students were permitted to test as often as necessary to achieve mastery. Time required for mastery was the dependent variable.

The independent variable was the presence or absence of Forced Correction of Errors in the practice items. Twenty practice items were available to the students, each a multiple-choice item requiring the student to classify a situation as describing one or another of the five coordinate concepts taught. As such, each of the practice items was one of a series of essentially similar items.

Students in both groups were told whether their responses were correct or incorrect, and received explanation of the correct answer. For those who erred, the explanation served to identify and justify the correct answer. For those who answered correctly, the explanation served to reinforce the response.

Students in the Forced Correction group were required to enter the correct response before proceeding. Students in the No Forced Correction group were permitted to proceed after receiving the explanation.

### Hypotheses

The research hypothesis is that time required for mastery will be significantly different (at the .05 level of probability) for the two treatment groups. Stated succinctly:

$H_R$: $a_{(1)} \neq a_{(2)}$

where "a" is the effect of the treatment in each group, (1) or (2).

The null hypothesis is the converse:

$H_0$: $a_{(1)} = a_{(2)}$

That is, the null hypothesis, to be tested by One-way Analysis of Variance, is that there will be no significant difference between the treatment groups.

### Sample

Subjects for the experiment were drawn from Instructional Technology classes at Wayne State University. It has been shown that prior verbal learning is crucial to attainment of defined concepts (Gagne, 1977). The present population can be assumed to have had successful experience at verbal learning tasks by virtue of their standing as graduate students.

The total sample was 43 subjects.

## FINDINGS

The following section deals first with the lesson as a whole, then with findings related to the hypotheses. The section concludes with data that relate to other measures that might be of significance.

### The Treatment Lesson

As an instructional lesson, the treatment was successful. Of the 41 students whose data were useful (two dropped out of the treatment during a PLATO system crash), 36 mastered the concepts. The breakdown of Mastery crosstabulated with the two treatments is presented in Table 1.

It is evident from the probability associated with this distribution that the treatment variables have no discernible effect on mastery.

### Findings Related to the Hypothesis

Table 2 relates the time required for mastery to the treatment. The differences appear minimal.

The One-way Analysis of Variance on the data in Table 2 is reported in Table 3. The difference is not significant at the .05 level.

### Discussion

The lack of significance found in Table 3 does not allow rejection of the hypothesis. Indeed, the difference in time required for mastery (from Table 2) is so slight that inclusion of the time required for the students in the Forced Correction group to re-enter an answer eliminates the difference entirely.

## Other Measures

In addition to the measure of time required for mastery, the lesson automatically determined the time required to complete the lesson plus the time required for the test. This total time measure is in recognition that learning may take place while testing, and that the total time may reflect changes not apparent in measuring the instructional time alone. Table 4 lists this total time, for each treatment group.

As before, the differences appear slight. Table 5 reports a One-way Analysis of Variance on this variable.

Once again, the difference is not significant.

Table 6 reports a count of number of practice items seen, by treatment group, since that count is also an indicator of instructional variation. Recall that, like time, the number of items seen was under the control of the students.

As before, the differences are not great, an impression confirmed by the Analysis of Variance reported in Table 7. The difference is not significant at the .05 level.

## DISCUSSION

Using a variety of measures, it appears that Forced Correction of Errors has not been shown to be significantly different from No Forced Correction.

## RECOMMENDATIONS

For the designer of instruction to be delivered by computer, the implications are not immediately obvious. There is, we may say with confidence, no compelling reason to require a student to correct an incorrect response, at least in the limited situation tested here. Neither, it must be added, is there compelling evidence in favor of No Forced Correction.

One previous study (Domenech, 1974) found that Forced Correction was of value when detailed constructed responses were required. In that instance, it may be supposed that the act of typing the response provided practice in itself. That practice effect is not evident when the response consists of a single keystroke, as was the case in this study.

It is clear that additional research needs to be done on this point. One possible variable of interest would be learning outcome. For instance, Rule-governed behaviors might well be different in this respect than the coordinate concept task in this study. The use of the question in the instruction might be another variable. For instance, a question used to measure retention of a generality statement might be different in its requirement for Forced Correction than the practice items used here.

If it can be shown that, at least in certain instances, Forced Correction is not of value, then the Tutor response judging should be modified. Others have noted that the Tutor -arrow- is clumsy to use, and unpredictable in its effect when modified. A thorough redesign of the -arrow- should include provision for No Forced Correction. Other CAI systems, including Phoenix, provide easier alternatives than PLATO in this respect.

## SUMMARY

One prevalent assumption is that design of Computer-Assisted Instruction is akin to design of Programmed Instruction. This study opens that assumption to question.

It may well be the case that Computer-Assisted Instruction has its own unique, and as yet poorly understood, rules. Rapidly growing use of computers for instruction gives urgency to studies of CAI design, so the production of courseware doesn't far outpace the design skills available.

TABLE 1

Crosstabulation of Mastery/Nonmastery by Coordinate Concept Instructional Treatment

| No Forced Correction | Forced Correction | |
|---|---|---|
| 15 | 20 | 35 |
| 2 | 4 | 6 |
| 88.2% | 83.3% | 85.3% |

NOTE.--Corrected Chi-square = 1.000, df = 1, P > .05.

TABLE 2

Treatment Mean Times To Complete CAI Lesson on Reliability

| Treatment | Mean (Seconds) | Converted To Min:Sec |
|---|---|---|
| Forced correction | 1366.3999 | 22:46 |
| No forced correction | 1354.6555 | 22:35 |

TABLE 3

One-Way ANOVA of Time to Complete Lesson

| Source | d.f. | Sum of Squares | Mean Squares | F Ratio | F Probability |
|---|---|---|---|---|---|
| Between groups | 1 | 1178.3828 | 1178.3828 | 0.004 | $p > .05$ |
| Within groups | 33 | 9 885 614.0000 | 299 564.0000 | | |
| TOTAL | 34 | 9 886 792.0000 | | | |

NOTE.--n=35.

TABLE 4

Treatment Mean Times to Complete CAI Lesson and Test on Reliability

| Treatment | Mean (Seconds) | Converted to Min:Sec |
|---|---|---|
| Forced correction | 1977.6499 | 32:58 |
| No forced correction | 1757.7998 | 29:18 |

TABLE 5

One-Way ANOVA of Time to Complete Lesson and Test

| Source | d.f. | Sum of Squares | Mean Squares | F Ratio | F Probability |
|---|---|---|---|---|---|
| Between groups | 1 | 414 289.5063 | 414 289.5063 | 0.691 | $p > .05$ |
| Within groups | 33 | 19 797 619.0000 | 599 927.8125 | | |
| TOTAL | 34 | 20 211 904.0000 | | | |

NOTE.--n=35.

TABLE 6

Treatment Means of Practice Items
Seen in Reliability Lesson

| Treatment | Mean |
|---|---|
| Forced Correction | 8.4000 |
| No Forced Correction | 10.1333 |

TABLE 7

One-Way ANOVA of Practice Items Seen in Reliability Lesson

| Source | d.f. | Sum of Squares | Mean Square | F Ratio | F Probability |
|---|---|---|---|---|---|
| Between groups | 1 | 25.7525 | 25.7525 | 0.235 | $p > .05$ |
| Within groups | 33 | 3618.5313 | 109.6525 | | |
| TOTAL | 34 | 3644.2837 | | | |

NOTE.--n=35.

## LIST OF REFERENCES

Domenech, O.G. A comparison of flagging-of-errors, non-flagging-of-errors, and forced-correction-of-errors in computer-assisted instruction requiring detailed constructed responses (Doctoral dissertation, The Catholic University of America, 1974). Dissertation Abstracts International, 1974, 35 (University Microfilms No. 74-19502).

Markle, S. M. Good frames and bad: A grammar of frame writing (2nd ed.). New York: John Wiley and Sons, Inc., 1969.

Markle, S. M. Designs for instructional designers. Champaign, Illinois: Stipes Publishing Company, 1978.

Merrill, M. D., & Tennyson, R. D. Teaching concepts: An instructional design guide. Englewood Cliffs, N.J.: Educational Technology Publications, 1977.

Schaefges, T. M., Eisenberg, J. D., & Stock, R. A. Beginning TUTOR (2nd ed.). Savoy, Illinois: Courseware Applications, Inc., 1980.

Sherwood, P. A. Tutor. Minneapolis: Control Data Education Co., 1974.

Skinner, B. F. The technology of teaching. New York: Appleton-Century-Crofts, 1968.

Tennyson, R. D., & Park, O. C. The teaching of concepts: a review of instructional design research literature. Review of Educational Research, 1980, 50, 55-70.

# The Microcomputer, Mastery Learning, and the Changing Role of the Instructor

Chris Dimas, Director, Computer-Supported Instruction

City Colleges of Chicago

## ABSTRACT

The purpose of this article is to present a rationale for a change in the role of the instructor. The advent of the microcomputer appears to have brought about a positive change in the attitude of faculty toward computer-assisted instruction. Simultaneously, some students are beginning to assume greater control of their learning environment and feel comfortable learning in a more independent manner.

Computer-managed instruction, which incorporates the salient elements of mastery learning, provides an accoutability factor in the learning process that many students and faculty will find appealing. Many of the more mundane chores now performed by instructors can be performed by the computer. Therefore, instructors will have more time to help their students on an individual basis achieve content mastery and understand the ramifications of their newly acquired knowledge; counsel and motivate their students; and recognize that their students are people who are more than merely "consumers" of knowledge.

The new faculty role made possible by the microcomputer should result in students achieving content mastery in a more efficient and effective manner. It follows that if students achieve at a higher level, instructor job satisfaction will increase. In the final analysis, both students and instructors will profit as the learning environment becomes more humane.

The use of computers in education has been with us for about two decades; however, it has just been within the last four years with the advent of the microcomputer that the interest of the educational community in general has been aroused. Although there had been outstanding work by some educational leaders in the field of computer-assisted instruction, most were content to take a "wait and see" attitude; some because of a fear that they could be replaced by the computer and some no doubt thought it was just too cumbersome.

## SMALLER COMPUTERS MORE ACCESSIBLE

The microcomputer, having followed on the wake of more expensive and sometimes less powerful computers, seems to have awakened the educational community. The privacy afforded by the microcomputer, as well as the ability to have control over it no doubt is very appealing to many educators. However, this should not lessen the fact that the intellectual curiosity of the educator has been awakened. Those who would have allowed the last years of their tenure to pass without much change are beginning to examine the microcomputer, and question whether it could have an impact on student learning. Having grown up with television, many of the young faculty see the computer's place in education as part of a natural sequence in the electronic age, and therefore are not quite as intimidated by computers as are others.

# The Microcomputer, Mastery Learning, and the Changing Role of the Instructor - Chris Dimas

5

Whether computers reach their potential in the field of education remains to be seen. At one time, motion pictures, radio, and television all held great promise and hope for educating our students. Also, the "systems approach" which relied on the use of behavioral objectives, criterion-referenced tests, various media, learning packets, CAI, text books, and most important, the teacher acting as a learning facilitator, counselor, and resource person was in vogue for about ten years. Then the systems approach slowly slid into oblivion about seven years ago. No doubt the reason for its demise was that it was a comprehensive approach to instruction and too many ideas were being introduced at one time. It was simply too difficult to motivate educators to abandon a comfortable mode of operation and acquire all the skills needed to implement the systems approach. Although eager to implement the systems approach, many administrators did not fully understand the extent to which learning courseware and resources were necessary; nor did they understand the extent to which in-service training was necessary to take on the new faculty role. Most faculty could not picture themselves as doing much more than disseminating content. And paradoxically, most students were not ready to accept a role which required them to take more of the responsibility for the learning process on an individual basis, despite the fact that they continually clamored for more voice in the governance of their educational institutions.

## EXPERIENCE WITH MAINFRAMES

Despite the problems inherent in the use of new educational techniques, we are here to examine the role of microcomputers and Mastery Learning. Old wine in new bottles? I don't think so. Many changes have taken place in the past eight years, and these changes have begun to influence the way both students and educators view the learning process.

For example, the educational community has the benefit of over twenty years of experience using large or mainframe computers for instruction. At City Colleges of Chicago, we have found that students who used PLATO (a mainframe computer-assisted instructional system) received higher grades than students who did not, and the more students used it, the greater their achievement (1981). Furthermore, in studies that we have conducted over the past seven years, 74% feel that they do better in classes that use PLATO to provide tutoring (obviously, there is no discrepancy between student perception of achievement and their actual achievement). Also, 90% of the students have consistently indicated that they liked to use PLATO to learn because it lets them take part at each step of the lesson. Students also indicated that they liked being able to learn at their own pace and liked avoiding the embarrassment of making mistakes in front of a group.

## TYPES OF COMPUTERS

Before proceeding, I would like to briefly discuss the types of computers in order to provide an overview of the field and also a perspective of how rapidly the field has and continues to evolve.

The three categories of computers are the mainframe, mini, and micro. Mainframe refers to the large computers such as the IBM 370 and the CDC Cyber-73, which are capable of supporting 60 to 625 terminals at one time. These computers are extremely expensive, usually costing millions of dollars, and requiring highly qualified staff to operate them. The power of these computers allows them to support sophisticated languages such as APL, PL/I, Smalltalk, and TUTOR, which in turn makes them easier to program and more able to provide sophisticated lessons. The mainframe at the University of Illinois presently supports approximately 1200 terminals, of which 625 may be active at one time. These terminals are supported on a time-sharing basis. For example, on the PLATO system, once every second or less, a terminal is "visited" by the main computer, and all of the necessary functions are performed within two hundredths of a second. All of the active terminals are visited before the computer returns to "visit" that same terminal.

The minicomputer came on the scene during the late sixties. This computer typically has from 64,000 to 500,000 words of memory, and can usually support from 4 to 32 terminals. Examples of the minicomputer are the PDP 11/44, produced by DEC, and the Motorola 68000. The low cost has made it relatively easy for colleges to own a minicomputer; however, the cost, which runs from $20,000 to $150,000 still makes it rather unattractive for personal use at home.

The microcomputer made its debut in education in the late seventies. These computers take up about as much room as a standard typewriter, and generally have from 16,000 to 48,000 words of memory. However, the use of floppy disks permit lessons to be easily and quickly loaded into the computer's memory. The cost of a microcomputer can be as low as $399; however, even the most casual user, after a few months, realizes that this basic unit is inadequate and that it is necessary to add additional memory, disk drives, and peripheral devices in order to accomplish a simple programming task.

## COMMUNICATION BETWEEN COMPUTERS

In addition, something new is almost upon us: the linking of mainframe, mini, and microcomputers by phone line in order for them to communicate with each other. For example, CDC has developed a stand-alone PLATO-type terminal (which can be termed a "micro") with disk drives, that can also communicate with a large mainframe computer. The mainframe can load a lesson into the micro in minutes, or a disk can be loaded manually.

Most of us in education have been deluged this past year with book salesmen from publishing houses who have turned into "computer experts", in order to sell microcomputers as well as books. One of the statements being made is that the microcomputer can easily communicate with any number of computers throughout the country via telephone line for the purpose of obtaining lessons. Although this is possible in some cases, at present it is not practical because of incompatibility in hardware and software. Often, when the languages are compatible, there are no lessons.

In the future, however, microcomputers will communicate regularly with other computers when enough lessons are developed and made ready for distribution, and as the languages become machine-compatible. No doubt a large mainframe will have the primary task of downloading these lessons to the microcomputers, or of providing a time-sharing capability to on-line users. Phone-line costs will probably dictate that users purchase disk drives in order that the mainframe can download to the microcomputer, thus saving phone line costs. If a major publisher obtains enough lessons, we can expect it to operate a commercial venture whereby microcomputer users can communicate with the publisher's mainframe.

Clearly, the limitations of mini-and microcomputers will diminish as time goes on. People will eventually be unconcerned with whether they are communicating with a main-frame, a mini, or a microcomputer, but whether they have immediate access to a particular lesson or program.

## WAYS TO USE COMPUTERS

Although there seems to be no single best way to use computers for instruction, the following is a summary of the main ways computers are used for instruction.

Tutorial: Students are taught a unit of study entirely by the computer.

Drill and Practice: Students are taught a unit of study by the teacher; however, the computer provides drill material until the student reaches a certain level of proficiency.

Simulation: Students having been taught the fundamentals of a particular subject can apply this knowledge in a life-like situation on the computer, where they can gain experience yet avoid experiences where the risk is too great, the time span too long, or the logistics insurmountable. By using computer simulations, the experience necessary to make wise decisions is gained prior to experiencing a real-life situation. For example, a medical student or intern is provided with a patient's symptoms, and is asked to make a diagnosis and prescription. After each prescription has been entered, the computer evaluates the patient and provides the intern with feedback on the patient's progress. Another example is the simulation of an experiment such as the mating of fruit flies in order to determine genetic characteristics, an experiment which would normally take a great deal of time to complete in the laboratory. Such simulations compress time, and aid the student in focusing on the intended outcomes as opposed to extraneous procedures. Finally, simulations allow for manipulating different variables in such areas as population growth. Within seconds, the student is able to project the need for schools and hospitals by manipulating such variables as the infant birth and mortality rate in a given area.

**Computer-Managed Instruction (CMI):** Students are directed to various materials and activities because of their performance on a pre-test on a course or unit. After studying the material, the student takes a post-test on the computer which is then graded and stored. If the criterion level is reached, the student progresses to the next unit; if not, he is provided an educational corrective (or prescription) via the computer. This process is usually repeated until the criterion level of achievement is reached.

**Computer Programming:** Students are taught how to write programs in order to solve problems encountered in the curriculum. For example, at Dartmouth College, all entering students are taught how to program in BASIC, and they are expected to use this skill throughout their college career.

Since it is clear that low-cost microcomputers can offer a variety of services to the educator, the question is what are the ramifications for Mastery Learning? The most obvious are those uses which have been made with mainframe and minicomputers. CMI has been used effectively for about the past fifteen years and has most of the same ingredients as Mastery Learning, including course and unit objectives as well as criterion-referenced quizzes and correctives. Combined with the microcomputer, CMI can free the teacher from performing some of the time-consuming and repetitious functions like administering, grading and recording unit and summative quizzes, and guiding the student to corrective material based on student performance. Mastery Learning advocates can surely benefit by becoming familiar wth CMI.

If college faculty must constantly perform repetitious acts involved in Mastery Learning, it is doubtful whether they will have the time to perform acts of humaneness for their students such as helping them on an individual basis to achieve content mastery and understand the ramifications of their newly acquired knowledge; counseling and motivating them; and finally, recognizing that they are people who are more than "consumers" of knowledge.

Recognition of a student as an individual by a teacher will enhance his/her self-concept at least while in the presence of that particular teacher. When the teacher also provides the student with content that has been carefully programmed into manageable steps leading to mastery, an even greater student self-concept will evolve. The combination of positive faculty interaction with individual students, along with student mastery of units of study, will result in an ever-ascending cycle of achievement, both on an academic and personal level.

## ACADEMIC SELF-CONCEPT

The self-concept is a prime determiner of whatever a person will accomplish (Wylie, 1961). Self-concept can be taught, and whatever can be taught falls under the domain of educational institutions (Combs, 1962). Faculty can perform those creative actions which will result in greater student achievement and positive self-concept, providing they are freed from such mundane duties which include administering, grading, and recording tests. The power to provide such freedom is here, but it remains to be seen whether the academic community will grasp the opportunity.

## ONE POSSIBLE FUTURE FOR CAI

Ten years ago, I wrote a report describing education ten years in the future. In this paper, I described a fictitious instructor, Mr. Smith, beginning his day by walking to the learning center and picking up the computerized progress reports on his students. After evaluating the reports, he met briefly with each student in the learning center in order to plan how each remaining objective in a unit would be reached. Students would then utilize videotape, programmed learning, learning modules, large or small group instruction, tutoring, or CAI in order to reach those objectives. After working with individuals on the learning strategy, Mr. Smith spent additional time leading group discussions in order to discuss the possible application of the new knowledge to real-life situations. The instructor then spent time following up on certain individuals by means of telephone calls and individual conferences. Finally, the day closed with an instructional design meeting where department members carefully worked out several paths which would help students with several different cognitive styles reach unit objectives.

Although I thought this futuristic report was rather unique, I soon discovered that many others had written similar reports. However, none of this ever came true except in isolated instances of modest proportions, and usually only when supported by government funding.

This opportunity for educational alternatives will probably not be revived or supported by government funding in the near future. Moreover, where the possibility for a learning revolution was more in the hands of the faculty and administrators, the next possibility will be in the hands of the student. It remains to be seen whether students will take the initiative for their own learning.

Gerald Gleason (1981) reported that the Carnegie-Mellon Institute projected that by 1985, every student will have a microcomputer. Today, publishers are beginning to develop whole courses of study for computers, supported by related materials. In the past, publishers sold books, and it was not necessary to validate the instructional capability of a textbook. With the advent of CAI, educators are asking more questions about the validity of particular instructional techniques and material. If publishers can state, for example, that a CAI course can bring about desired results, it is quite likely that a percentage of students will simply buy the course and take it at home. When they have finished the course, they will earn college credit by taking something equivalent to a CLEP test or by obtaining credit through life experiences. I do not mean to say that the classroom is in jeopardy. Far from it. However, students with unusual schedules and learning styles as well as the non-classroom oriented student will probably find this mode of education appealing.

From the faculty point of view, the teaching/learning process in the age of computers should become more rewarding. Presently, teacher "burnout" is reaching epidemic proportions. And why not? Instructors are human beings with all their frailties. After years of repeating the same content, it is quite possible that some individual instructors may almost begin to view themselves as machines and become negative toward themselves and their students. This is especially true in classes with slow learners. Students who do not learn pose an ego threat to faculty. After all, how can one be called a teacher if the students are not learning?

## HUMANIZING THE TEACHING PROCESS

The microcomputer will humanize the teaching-learning environment, by permitting the faculty to perform those functions which can only be performed by human beings, while delegating functions which are repetitive and mechanistic to the microcomputer. All faculty will be allowed to become as creative as they wish in the teaching-learning process, and in so doing, experience greater job satisfaction. In return, we can expect greater achievement from students not only because of the changing faculty role, but also because the various paths to content mastery have been well marked, with many vehicles for its achievement.

1) Combs, Arthur W. (ed.), Association for Supervision and Curriculum Development Yearbook of 1962, Perceiving, Behaving, Becoming. Washington, D. C., 1962.

2) Dimas, Chris, "The Effect of PLATO on Student Achievement and Attitude", 1981 Conference Proceedings: Association for the Development of Computer-Based Instructional Systems, Western Washington University, Bellingham, Washington.

3) Gleason, Gerald J., "Microcomputers in Education: The State of the Art", Educational Technology, March 1981.

4) Wylie, Ruth E., The Self-Concept: A Critical Survey of Pertinent Research Literature. Lincoln, Nebraska, University of Nebraska Press, 1961.

# AN ANATOMY SELF-TESTING PROGRAM
# ON PLATO

K.M. Bagnall*, M. Szabo+, R. Frost+, and I. Rollingher*

University of Alberta

## ABSTRACT

A lesson has been developed which allows students access to a critical and informative "professor" at all times and which also allows access to anatomical situations outside of the regulated periods in the dissecting room. The purpose of this lesson is to permit individual students studying anatomy to be tested on their knowledge in a highly individualized fashion with minimum professorial involvement.

This is a first step towards producing a model of a professor of Anatomy who is available to all students at any time. Primarily, there is the facility for a student to simulate a comprehensive examination situation. In its present form the lesson is used in the Departments of Anatomy (Faculty of Medicine) and Faculties of Nursing, and Physical Education and Recreation. The flexibility of the lesson lends itself well to other disciplines.

In 1980 Mah studied several features of PLATO which could be utilized in the teaching of Anatomy particularly to first year medical students during their basic science training. Her conclusions were that:

1. question and answer with feedback
2. touch screen
3. graphics with REVOX
4. instructional formats

lend themselves well to the teaching of this practical subject. In 1981 Rollingher undertook a project based on these conclusions to develop a PLATO lesson which approached the characteristics of a professor of Anatomy. Such a lesson would allow students access to anatomical situations outside of the regulated periods of the dissecting room.

The major features of the lesson that was developed are:

1. The anatomical areas of study are presented in a regional fashion which the student can select. PLATO then draws at random appropriate questions from the data bank for that area. Options are available for the addition and/or expansion of a region or regions as well as the division of an area into smaller sub-regions.
2. There are four types of questions possible. The multiple-choice, true/false and type K are of a standard nature. The fourth type is a combination of the other three and perhaps the most exciting. It makes use of a REVOX machine coupled to a PLATO terminal. The random access REVOX will project up to sixty photographs of specimens previously mounted on a microfiche card. The questions that can be asked are of the format of the other three above but are related to a photograph of an actual specimen. This is vital for a visual subject such as Anatomy.
3. For each area of content selected the student is able to choose the type of question, the number of questions desired and the length of time required to answer each question. He/she also has the choice of choosing a time to complete a selected area. Quite clearly, there is ample flexibility here to create a situation that closely resembles an examination.
4. The screen on a PLATO terminal is touch-sensitive and the lesson utilizes this feature extensively. Any input required from the student can be entered by touching the appropriate area on the screen. The student

# CAI IN FOREIGN LANGUAGE TEACHING:

## COMPONENTS OF A MICROCOMPUTER STATION

Robert A. Ariew

The Pennsylvania State University

One of the most significant advances in Computer Assisted Instruction in recent years is the availability of small, relatively inexpensive, microcomputers. These devices can duplicate the majority of the functions of even the most sophisticated CAI system at a fraction of the cost. However, there are some drawbacks in their capabilities. The microcomputers are not designed with CAI tasks in mind; that is, they are not specifically equipped with the hardware capabilities most sought in CAI development. Instead, they are designed to be "personal machines", flexible enough to serve in word processing, game playing, small business accounting, or in small-project scientific research, but exactly suiting none of these tasks. While it is true that the micros can be configured to accomodate many tasks, the process is not entirely painless. It is often necessary to deal with several different software and hardware companies to properly equip a CAI station. And then of course, the price per station escalates as one adds capabilities, negating in an appreciable way the benefits of using microcomputers in CAI.

Modern foreign language teaching involves the acquisition of several skills simultaneously, including listening comprehension, speaking, reading and writing. Traditionally the foreign language class has stressed reading and writing. However, today one cannot ignore speaking and listening comprehension skills since oral communication is now considered important in the foreign language class. Therefore, to teach foreign languages with microcomputers requires, in addition to the "normal" keyboard, video display and mass storage station, a means to produce audio cues and reinforcements. Otherwise listening comprehension cannot be dealt with at all. Since it is already practically impossible to teach speaking skills with a computer (computer voice recognition is still difficult to achieve), if one omits listening comprehension, computer assisted foreign language teaching becomes nothing more than a series of exercises dealing with written grammar. Such exercises could probably be done just as well on paper.

In addition to having a means to store and play-back audio information, the CAI station must have thecapability to display foreign language characters and diacritics (for some languages, Chinese for example, this is a very complex problem to solve). The student must also be able to produce the characters and diacritics easily from the keyboard.

Other capabilities, such as the use of color to draw attention to the explanations, the use of graphics for emphasis, the ability to show slides, the ability to enter answers through a light pen or a touch sensitive screen are all useful, but not essential for foreign language teaching.

The minimal configuration of a foreign language CAI station is: a keyboard and screen capable of accessing and displaying foreign language symbols, a mass storage device for lesson materials, and an audio device.

never has to touch the keyboard although this option is always available. This enables the focus to be entirely upon the screen.

5. Feedback is a vital element in any educational system. This lesson is designed to provide the student with as much feedback as possible:
   a) For every question the student is given feedback as to i) correct or incorrect (in a variety of expressions); ii) the number of questions attempted, the number to be attempted, the number correct so far; and iii) time taken for each question (run-down clock).
   b) The student has the option to skip a question and try it later (it is added at the end) or if he/she gets it wrong the first time to do it again. If he/she gets it wrong twice, the answer is provided.
   c) Provision has been made for the insertion of a comment after each question, giving the student some useful information directly relating to the question integrated with his answer.
   d) Between every 15-20 questions (random) the student receives feedback as to his progress relative to the class mean for the area he has selected. He also receives appropriate comments related to his performance on the previous 15-20 questions. If the student is continually failing, for example, he is removed from the lesson and strongly urged to sign-on again.
   e) A concept closely related to feedback and felt essential to the success of this lesson is that of human contact. Computer based instruction, being machine based, is subject to the trappings of mechanization with potential to quickly become cold and impersonal. This will not sustain a student's interest; it cannot stimulate him and the ability of the lesson to convey information is compromised. Attempts have been made to overcome this by:
      i. calling the student by name and recognising this each time the student uses the lesson - "Hi there (name). Good to see you again".
      ii. Random evaluations of his work with appropriate critical comments (25 levels of criticism are available) - commendations for good work, gentle prodding when required.
      iii. The facility to use "comments" directly to the author from the lesson.
   f) The student data collection is extensive. Data collected include:
      1. Marks for each content area for each student.
      2. Marks for each occasion for each student (recording the last 8 sessions).
      3. Number of hours spent at terminal.
      4. Breakdown of marks for each occasion.
      5. Evaluation of each question.

      These data provide continuous output that monitors both student progress and the efficiency of the lesson and the questions asked.
   g) The anatomy self-testing program is highly flexible as a piece of computer software and can readily be applied to other diciplines. It is capable of expanding its data bank by adding more questions and/or regions. It can also accomodate new users or new types of users. At present time students from various departments, notably Dentistry, Nursing, Rehabilitative Medicine and Physical Education have begun to use the facility. Furthermore, other faculties are preparing their own question banks.
   h) A "question-editor" driver program has simplified the addition of questions to the lesson. The driver presents the question formats in template form, requiring only blank spaces to be filled in as the entering procedure. This requires no knowledge of TUTOR language whatsoever allowing a secretary with no extraordinary computer skills to be employed.

At present approximately 1000 students per year have access to this program. Students' comments will be evaluated and any necessary improvements made to the lesson. In the meantime we feel that we have taken a first step towards producing an "out-of-hours" professor that is of considerable value to the students of Anatomy.

# INTEGRATED MUSIC PROJECT

W. K. Bales, K. N. Paoli, E. Weherman

Western Illinois University

A computer-assisted learning facility is under development at Western Illinois University (WIU). The Integrated Music Project (IMP) is concerned with those elements of musicianship which are common to all areas of music. Therefore, courseware support for music theory, music education, elementary guitar and other disciplines is being planned. While each lesson area will have its own format, each will reinforce concepts presented in others. Computer literacy will be emphasized in an effort to help produce students who are prepared to function effectively in our computer-based society.

Hardware for the IMP includes four Apple II microprocessors with 48K RAM, Apple language systems, pairs of disk drives with controllers, and ALF music synthesizers. Programs under development are written in Pascal, 6502 Assembler, and the North Texas State University Curriculum Driver with the AMUS score input language. Aural skills training for music fundamentals and basics of the guitar are currently being written, and it is hoped that limited student use of the system will be possible by the Spring semester, 1982.

This project is funded by the WIU Office of Faculty Development, the College of Fine Arts and the Department of Music.

# COMPUTER LITERACY: A SURVEY OF TEACHERS K THROUGH 12

Julie Bengtson

California State University at Chico

The increased use of computer applications in the last decade has increased an awareness by the scholastic community for computer literacy. To achieve computer literacy the first requirement must be knowledgeable, computer literate teachers.

Further, computer literacy brings attention to the immediate need for teacher training. In an age of advanced technologies, the learning experience must prepare students for work in a society which places greater emphasis on technical skills. Exposure to computers in every facet of our society, including the home, will give today's child the potential for becoming more computer literate than his instructor. With appropriate training, teachers at every level can meet this demand for technical knowledge.

My thesis for California State University at Chico is a survey designed to measure the attitude of teachers toward computers and their use in an educational setting. The questionnaire istargeted for a population of teachers in grades K through 12. It is my intention to determine if independent variables such as age, field of study, length of time since graduation from college, and additional training or education within the last five years will influence the outcome of my research. I will also be evaluating knowledge of general computer terminology.

Within the body of the paper a definition of literacy is provided in order to establish qualifications for computer literacy. Computer literacy is designated in four stages to refine levels of competency with respect to general knowledge, computer applications, and computer hardware. The one-page questionnaire will take 15 minutes to complete and is scheduled for distribution in May of this year.

# PROJECT U-TRAIN/CAL - THREE YEARS ON

MAJOR J.J.R. Bélanger, SO CAL

Lt(N) Griff Richards, SO CAL 2

## ABSTRACT

PROJECT U-TRAIN/CAL is an investigation by the Canadian Forces (CF) into the implications of Computer Assisted Learning for self-paced technical training. This report traces the CF involvement in CAL, the objectives of the present program, implementation considerations, courseware development and future developments.

## INTRODUCTION

The Canadian Forces train over 12,000 persons in over 120 trades per year. This training is achieved by formal coursing and/or on-job training. In achieving its training objectives, the Canadian Forces must contend with a multitude of problems which reduce training effectiveness. Problems which may be solved by altering the training system include: (Bélanger, 1979)

a. the near total dependence on instructors as the principal means of delivering instruction. The implications of continuing this approach are several. To name but a few:
   (1) a heavy recurrent need to qualify tradesmen as classroom instructors,
   (2) the distinct possibility that varying levels of instructional competence will continue, and
   (3) the continued inability to capitalize on varying rates of learning among trainees;

b. the considerable cost involved with pursuing training using large numbers of actual equipment and true fidelity simulators; and

c. operational positions remain vacant due to delays in training or extensive training times.

## SELF-PACING

During the last few years, the Canadian Forces has been seeking efficiencies in training through the use of educational technologies. One concept which has emerged as a successful innovation in training, is self-pacing. This does not imply that our internal training problems would all disappear in a non-instructor intensive environment, but rather many would be alleviated.

Before proceeding any further, it is important to understand what is meant by self-pacing.

> "Self-pacing allows trainees to learn, practise and review skills and theory independently at their own rate of speed."

The provision of an environment which provides for learning, practise and review covers the total circumstance. The inclusion of skills and theory in the definition underlines the fact that self-pacing is applicable to both types of training. Although, the trainee proceeds independently he is not without an instructor; one-to-one contact is an integral part of the self-pacing concept especially when a trainee is having difficulty. The classic role of tutor is the basic foundation of the instructional setting and self-pacing fosters this. The other side of the ability to proceed independently is the capability of the learning package to "stand alone". The package must contain everything necessary to allow trainees to learn on their own. Our definition concludes with the important concept that trainees proceed at their own rate of speed and it is here that the pay-off is most evident. The fact that many, even the majority, would finish the training early and therefore be out of the Training

System and into the operational commands is a major benefit to the Canadian Forces. Those trainees who are moving slowly will have the advantage of extra assistance and therefore the likelihood of training failures is reduced. (Sapp, 1977)

It was for these reasons that the Commandant of Canadian Forces Training System (CFTS) authorized Project U-Train, a study of self-pacing, in November 1977. The aim of the study was to determine if self-pacing had relevance and application to Canadian Forces training.

Many different media are available for self-pacing courses. They include paper programs, video tapes, sound and slide, and computers. Of all existing media, computers have the greatest potential in power and versatility to deliver instructions as it can present information through its own delivery system or by controlling secondary media i.e. video disk, 35mm slides, audio tapes.

The unique potential of the computer to deliver instruction lies in three areas. They include tutorial, drill and practise, and modelling and simulation. The computer can also be used to manage training by marking and analysing tests, maintaining student records and, in an individualized training scheme, prescribing lessons according to a student's performance. For these reasons, Project U-Train was expanded to include Computer Assisted Learning (CAL) as one of its components in November 1978.

## PROJECT U-TRAIN/CAL

## U-TRAIN/CAL - HISTORICAL

The Canadian Forces began its investigation into the possible use of computers as a medium of instruction in 1972. From April '75 to July '77 the Canadian Forces undertook a small scale CAL experiment called the Educational Technology Project. The project was conducted at the Canadian Forces School of Communications and Electronics (CFSCE). The aim was to determine the feasibility of CAL as a medium of instruction. The project demonstrated that CAL was feasible for Canadian Forces training. (Petruk, 1977)

Unfortunately no plans had been made to exploit the project's success and the program was suspended.

In November 1978, 426 (Transport) Training Squadron - the CF school responsible for all aircrew and technical training on fixed wing transport aircraft - expressed an interest in and agreed to undertake a new small scale CAL test. This provided CFTS with the potential to evaluate CAL for Project U-Train.

The test consisted of converting, conducting, evaluating and validating one course. The course selected was the Gas Turbine Compressor for the CC130E (HERCULES). The CAL courseware was programmed in NATAL by an Aero Engine Technician who was a 426 (T) Trg Sqn instructor. This instructor had never programmed before. This CAL course offered time reductions of 30%-60% over classroom instruction. Encouraged by these results, 426 (T) Trg Sqn asked CFTS to expand the scope of the test.

CFTS and 426 (T) Sqn with Air Command approval agreed to expand the scope of the study by developing an additional five CAL courses. The courses selected were:

a. solid state devices;
b. T56 engine, fuel and propeller systems;
c. Boeing 707, air conditioning and pressurization system (later suspended due to personnel transfers);
d. T56 engine syncrophaser; and
e. CC130E aircraft, Crash Position Indicator (CPI) and Flight Data Recorder (FDR).

A sixth course, Digital Computer Fundamentals, was later added to the list by CFTSHQ. This course would be developed by CFTS/CFSCE and Canadian Forces Fleet School Halifax (CFSS(H)). Encouraged by the program expansion, National Research Council selected CFTS/426 (T) Trg Sqn as the test school for its prototype NATAL mini-system.

In order to meet our obligation to the National Research Council, additional funding was sought within the Department of National Defence. The department's Chief of Research and Development (CRAD) provided the additional funds.

## U-TRAIN/CAL OBJECTIVES

CFTS primary objective is to evaluate CAL as a component of Project U-Train. Sub-objectives are:

a. gaining experience in training authors/programmers in preparing self-paced courses and programming in NATAL;
b. determining the potential of the computer to simulate equipment faults and to teach fault-finding/troubleshooting techniques using hardware free (synthetic) simulators;
c. helping 426 (T) Trg Sqn implement a series of technical CAL courses using state-of-the-art technology;
d. developing a plan for the expansion of CAL in the Canadian Forces using, wherever possible, existing departmental resources;
e. determining the computer configuration required for an effect-

ive CAL delivery system;

- f. evaluating the cost of exporting the courseware to the trainee via remote terminals;
- g. determining the requirements for a Computer Managed Learning system; and
- h. determining the organizational and personnel establishment required by a CAL system.

426 (T) Trg Sqn's primary objective is to determine the effectiveness and efficiency of CAL to aircraft maintenance training. Sub-objectives are:

- a. to assess:
  - (1) CAL's capability as a training delivery medium in the aircraft maintenance field,
  - (2) CAL resource and scheduling implications,
  - (3) CAL's effect on reducing or increasing recurring training costs and resource requirements,
  - (4) the practicality of remoting the training to the flight lines;
- b. to evaluate the benefits of CAL training by identifying and tracking selected maintenance costs;
- c. to evaluate the use of maintenance instructors who are Subject Matter Experts (SME) as CAL authors/programmers;
- d. to determine the optimum student environment; and
- e. to determine the optimum student administration method.

### U-TRAIN/CAL - IMPLEMENTATION

### Training the Instructors

Subsequent to the selection of the course, steps were undertaken to train the selected instructors in the preparation of self-paced materials (Instructional Material Design (IMD)) and NATAL authoring as most of the instructors were untrained in either discipline.

The IMD course consisted of the following:

- a. the individual training system:
  - (1) job analysis,
  - (2) task analysis,
  - (3) performance objectives and criterion tests,
  - (4) training objectives and suggested methods and media;
- b. programmed instruction techniques:
  - (1) techniques of branching, constructed response, selected response and adjunct programming,
  - (2) editing and evaluation of training programs are also taught;
- c. application and feedback.

The NATAL programming course prime objective was to provide the instructors with adequate knowledge to program in NATAL. The average time to train the instructors was 2 weeks.

NATAL has gained widespread acceptance with the instructors due to the following:

- a. the ease of learning which has permitted them to concentrate on the functional aspects of their course;
- b. the high level of interactivity with different, but functionally equivalent devices; and
- c. NATAL's modularity in developing courseware.

### Support Training

The acquisition of a computer system normally requires vendor or equipment specific training. A CAL system is no different.

A conscious decision was made to train some instructors as computer operators and others for first level terminal repair. The decision to do our own first level terminal repair was imposed by the location of the terminal suppliers - Lektromedia in Carleton Place and Volker-Craig in Kitchener-Waterloo.

One member of the project will be trained as the system manager/programmer. This person is the only one having an in-depth computer background.

### Learning Centre

The successful implementation of CAL is very dependent on an appropriate learning environment.

To meet this need, two classrooms at 426 (T) Trg Sqn were converted. Physical renovations included changing the lighting fixtures, lowering the ceiling, increasing the number of electrical outlets, soundproofing the rooms - i.e. installing anti-static carpets, ceiling tiles, wall tiles and dividers - and painting.

The final stage consisted of designing a student work station as none of the commercially available ones met our requirements. Many designs were proposed by members of CFTS and 426 (T) Trg Sqn before final selection.

### Courseware Development

426 (T) Trg Sqn provides 10 Subject Matter Experts of various trades and rank. As their primary duty is traditional instruction, the CAL Courseware production is done on a time available basis. 426 (T) Trg Sqn has also made available one officer with some computer training as their project coordinator. The only course being developed with full-time authors/programmers is the course on Digital

Computer Fundamentals. This explains why even though design of this course began only in February 1981 and is the longest conventional course, it has seen the fastest development (Table 1).

As the SMEs are not instructional technologist or programmers, CFTS has made available one full-time training developer from its Training Development Centre and one CAL/ADP specialist from its Training Development Division. CFFS(H) has also made available one instructional technologist and an instructor to assist in developing the Digital Computer Fundamentals Course. No ADP specialist is required at CFFS(H) as the author is an experienced programmer.

Table 1 shows the time spent on each course since April 1979.

A detailed description of the authoring methods used in developing the courseware is found in the article by Richards and Bélanger. (Bélanger and Richards, 1982)

### Lessons Learned (Experience Speaking)

The first course implemented in NATAL-74, the Gas Turbine Compressor (GTC) for the CC130, taught us several lessons.

The first was that even NATAL with all its "transportability features" was to some degree terminal limited with respect to touch panel responses. The GTC was developed on Lek120 terminals - 24 rows, 80 columns - while the new terminals Lek304 - 32 rows, 80 columns - required the conversion of all touch response boxes. It was decided that this problem could be alleviated by changing the NATAL CG (row, col, nrow, ncol) function to NCG (row, col, nrow, ncol, X, Y) function where:

row, col: upper left corner of box
nrow: number of row from top to bottom
ncol: number of columns from left to right
X, Y: author specific grid independent of terminal

The function NCG ( ) takes the authors grid and translates it into the terminal absolute gird, thereby making the touch responses universal.

The second lesson learned was that an examination data bank for every course should be available independently of the course. DND has started to write a utility program that would allow any author to create these questions and then assembled to form tests, exercises or final exams. This utility program is being written by a Master Corporal, who until he learned NATAL had never programmed before. This program should be ready for use by the end of Jun 82.

## U-TRAIN/CAL - FUTURE DEVELOPMENTS

### Synthetic Simulator

The troubleshooting of complicated, expensive, delicate equipment that malfunctions and must be made serviceable as soon as possible is similar to the skills of a physician who must deduce a differential diagnosis of a "human malfunction" before his patient dies. For this reason it is essential that the technician of the '80s be given more effective training for the learning of his trade. The synthetic or hardware free troubleshooting simulator is regarded as the key to this improved training. Some advantages of such a simulator are:

a. it can be easily modified;
b. it can teach a system approach to troubleshooting; and
c. it is safer than the real equipment when performing dangerous procedures.

CFTS has a prototype version of such a simulator presently available for restricted use.

### Remote Coursing

A major advantage of CAL is that the identical material offered in the school could be offered to a trainee at his place of employment if the appropriate resources are made available. Some advantages of this approach include:

a. reduction in travel costs and time;
b. reduction in time away from work; and
c. availability of courseware.

### Instructional Support Systems (ISS)

A successful CAL system must not only provide the basis for instructional interaction, but it must also provide administrative support and monitoring functions related directly to the support of instruction. NATAL provides for most of these ISSs. Two areas where NATAL presently provides no ISSs are in the training of CAL courseware authors and NATAL programmers.

### SUMMARY

Project U-Train/CAL is still in the courseware design and authoring stage. The success achieved to date with instructor SME as instructional material designers and NATAL programmers bodes well for the success of this prototype project.

(1) Statement of Requirement for a Computer Assisted Learning Pilot Project to be Conducted Under the Auspices of CFTS Project U-Train, Major J.J.R. Bélanger, 29 Oct 79
(2) Project U-Train, A Concept of Self-Pacing, Major R.D. Sapp, Nov 77
(3) Final Report on the Canadian Forces Educational Technology Project Submitted to the National Research

Council of Canada, M.W. Petruk, Sep 77

(4) Authoring Methods for Subject Matter Experts Using NATAL-II, Major J. Bélanger and Lt(N) Richards, ADCIS 82, June 82

(5) Current Applications of Computer Assisted Learning in the Canadian Armed Forces, Capts J. Bélanger and A. Madsen, Sgt R. Drake, ACIT 80, Feb 80

## COURSEWARE DEVELOPMENT HOURS

(START DATE)
(TOTAL HOURS)

| COURSE | ANALYSIS | DESIGN | AUTHORING | DEBUGGING | TRIALS | CONVENTIAL COURSE LENGTH (TRG DAYS) |
|---|---|---|---|---|---|---|
| T56 Fuel | Apr 80<br>110 | May 80<br>366 | Sep 81<br>9 | | | 8 |
| T56 Prop | Apr 80<br>47 | May 80<br>182 | Apr 81<br>287 | | | |
| T56 syncro-phaser | Apr 80<br>140 | May 80<br>311.5 | Apr 81<br>380.5 | | | 5 |
| CC130 CPI | May 80<br>175 | Jun 80<br>207 | May 81<br>315 | | | 4 |
| CC130 FDR | Jun 80<br>228 | | | | | |
| Boeing 707 Air Conditioning | May 80<br>174 | (Suspended) | | | | 5 |
| Solid State Devices (TSQ:TB) | May 80<br>317 | Mar 81<br>306.5 | | | | 10 |
| Digital Computer (TSQ:XH) | Apr 80<br>528.5 | Feb 81<br>1458 | Aug 81<br>1640 | Jun 80<br>83* | Jun 80<br>16* | 30 |

TABLE 1

* Sound slide introduction package
All times as of 31 Jan 82

# PROJECT REPORT

Lisa P. Brenner

Rush University

Rush University is a part of Chicago's Rush Presbyterian St. Luke's Medical Center, with an enrollment of approximately 1,000 students in the College of Medicine, Nursing, Health Sciences and Graduate Studies. The Rush model of education emphasizes the role of teacher/practitioner. Aside from their academic responsibilities, faculty have important service and research roles within the institution. They are consequently eager to optimize their academic responsibilities.

After experimentation with IBM's Coursewriter, mini-computer simulations, and micro development, Rush connected to the Urbana PLATO system. Since PLATO was installed in November of 1979, the number of terminals has expanded from four to thirteen, and instructional use has grown from 120 hours per year to 2,000 hours per month. With a library of over 400 PLATO health science lessons, a manageable number of students, administrative support, and a faculty which is not concerned about being replaced by computers, Rush is an ideal environment to test the degree to which computers can be incorporated in health science education.

The availability of a large library of existing PLATO health science courseware has been an invaluable resource for orientations, assignments, and as development models. By spring of 1982, we had given 120 faculty and 500 student demonstrations. Forty-eight faculty had participated in development projects, and 44 faculty had assigned PLATO lessons to students.

The Office of Computer Based Education supports the development of instructional programs which will be assigned to Rush classes. Twenty-two programs have been developed for use at Rush, constituting about 20 hours of instruction. Programs are generally developed by a faculty content expert and a student programmer along with staff consultation. We train student programmers, who then provide not only programming support, but instructional feedback and advertising to other students and faculty about the virtues of computer based education. Both students and faculty are eager to establish item banks for testing and review purposes. Banks in pharmacology, physiology, Biochemistry, microbiology, behavioral sciences and medial technology have been developed.

The Rush Guide to PLATO Instructional Development contains our forms, procedures, suggested principles of design, and "lesson essentials" which all new programs must meet before assignment to students. Besides individual faculty and new student demonstrations, we offer periodic workshops and continuing education programs for faculty and staff interested in PLATO use and development.

We have only begun the process of incorporating computers into the educational system at Rush. The most delicate problem has been gaining enough faculty committment to justify expansion without overtaxing existing resources. Faculty must become familiar with existing materials and experience positive feedback from students before they will become regular users. Providing these experiences with four terminals to classes with over 100 students was not uniformly successful. On the other hand, if faculty are limited to assigning supplementary computer based education lessons, they do not see the time or cost savings that accompany computer based education. Other experiences that will be included in the project report are: The extent of faculty involvement, pro and con's of locating termings in the Learning Resource Center, controlling the computer hobbyist contingency and persuading faculty to substitute PLATO for lectures and other assignments.

# MICROCOMPUTERIZED TACHISTOSCOPE TO IMPROVE SKILLS IN ALGEBRA (PROJECT REPORT)

James B. Caballero - University of California, Los Angeles

Students in precalculus courses at the University are often hindered by their poor performance with fundamental skills in algebra, supposedly acquired in a previous course. Just as a tachistoscope can be used to improve reading skills, mathematical skills can also be improved. An Apple II computer is used to generate problems which appear on monitors for brief intervals, usually less than one second. The class responds with an answer, which is then displayed for the class. Effectiveness of this technique is measured qualitatively by student performance on examinations.

This informal study is in its preliminary phase, that is, qualitative data is being collected by comparing student performance on tests which are distinguished by their length. Students in classes which are exposed to this treatment twice a week for a maximum of five minutes per exposure are capable of excellent scores on examinations which are 50 to 100% longer than those used in untreated classes.

The problems used in precalculus are those from first year (high school) algebra. Typical problems are solution of simple linear equations, factoring trinomials including special forms such as difference of squares, greatest common divisor, and reduction of fractions. The computer generates problems within constraints imposed by the programmer. Timing and program sequencing are controlled by the teacher using the game paddle.

Earlier investigation has shown that some students can solve a linear equation with unknowns on both sides after an exposure of 1/50 seconds. Two important observations have been made. Not only does the subject quickly learn to function with a briefer exposure but the interval between exposure and response is similarly reduced. Secondly, when exposed to this treatment for skills where concepts have already been learned, the mental processing is frequently at a subconscious level. That is, a correct answer can be obtained without conscious knowledge of the actual processing which led to that solution. It is as though the mind is much more alert as a result of the anticipated brief exposure.

The benefits to be derived by such a treatment seem to outweigh the relatively small amount of time devoted each week to the program.

# BARNEY: A COMPUTER WHO CARES ABOUT ADOLESCENTS  HEALTH

B. Chewning, D. Gustafson, K. Bosworth, T. Day, R. Hawkins

University of Wisconsin-Madison

The Body Awareness Resource Network (BARN), known as Barny to its users, is a computer based health information system for adolescents, ages 11-18, being developed and pilot-tested by the Center for Health Systems Research and Analysis (CHSRA), under a Kellogg Foundation grant. The overall goal of the BARN Project is to promote adolescent health by helping adolescents, 1) maintain positive health habits, and 2) evaluate and change negative health habits.

The BARN system has programs on five topics: alcohol and other drug abuse, body management (diet and exercise), sexuality, smoking, and stress management. Across each of the programs BARNY, a non-judgemental, compassionate, anti-hero does the following: 1) offers adolescents concrete information about consequences of negative vs. positive health behaviors, 2) offers strategies for dealing with health concerns in these areas, 3) helps adolescents build skills for resisting peer pressure, 4) suggests what one can (and cannot) do for friends who are taking risks, 5) expands the user's skills for anticipating and avoiding situations involving personal health risks, 6) describes community resources, and 7) encourages adolescents to identify and draw upon their natural support network (e.g., family and friends).

The programs have been designed for viewing by an individual adolescent in one of three settings: 1) public schools, 2) clinics, and 3) public libraries. Privacy, confidentiality and a non-judgemental tone have been considered critical for BARNY to be effective.

A fourth site has been added for group viewing. As the health behavior literatue attests, it is difficult to influence or change health behaviors, particularly in the absence of strong social support. For this reason 4-H clubs in two Wisconsin counties will use the programs to encourage peer support and also to involve the adolescent's family through viewing.

In Phase I, Program Development (June 1981 - August 1982), each BARN program is being developed based on literature, and input from adolescents, parents, service providers and researchers on the program topic. Programs are being pilot-tested in clinics, middle and high school settings. In spring 1982, a pre-test questionnaire on health knowledge, attitudes, behaviors will be administered to adolescents in the control and experimental school sites. In Phase II, Implementation (September 1982-June 1984), the BARN system is to be installed in the school, clinic and library experimental sites.

The post-test questionnaire will be administered in the control and experimental sites in the fall of 1983.

# USE OF VIDEO TAPE TO PRESENT AN INTRODUCTION TO COMPUTING

Alex Rees Clark

Temple University Computer Activity

The labor intensiveness of providing user training can be reduced by video tape presentation of topical training materials and general introductory subjects. This paper will describe a video taped general introduction to computing hardware and software.

People encounter computers in several ways. They may take a programming course or use computers in their jobs. Such introductions do not provide users with an overall view of how computers function and how systems are organized. Later, a user's changing needs may create a heavy load on consulting services. This consulting load can be reduced by providing appropriate user training.

At Temple, a comprehensive series of user training seminars is provided each semester. Among these for several semesters has been a one-session, two-hour seminar entitled "everything you Always Wanted to Know about Computers, but Were Afraid to Ask". This topic's breadth required that it be taught by a relatively senior member of the consulting staff. Each two-hour session required approximately one hour to prepare. The seminar was offered approximately six times per semester, and that number was growing steadily, pushing the cost per semester into the thousands of dollars.

The video tape has several advantages over the lecture format. The standardized seminar can be presented by a junior consultant. Moreover, the tape can be made available through a media center for individual or group showing without a consultant/presenter. In the case of our tape which we call "Computers and You", our lecture presentation which ran two hours was compressed to approximately twenty-five minutes of video tape. The video tape is now presented in a one-hour period with a hand-out that contains a glossary and some amplifications of points made in the tape.

Several lessons were learned in the preparation of the tape. First, one does not directly transfer an oral presentation to a video tape without making substantial changes in the organization of the information and the visual aides. Video allows for the juxtaposition of various verbal and visual information. It's possible, to show a real computer system in operation or to simulate sound. Second, the primary focus in the design of a video presentation must be on the script, which must be completely prepared in advance of any taping.

"Computers and You" was completed in January, 1982 and used in approximately a half-dozen presentations so we have not had much opportunity to evaluate its effect. We estimate that we have reduced the cost of presenting our seminar by approximately two-thirds. In addition, comments by those attending the session have been generally favorable. Persons attending have taken copious notes, and the level of questions being asked of the presenters has been higher than in the lecture mode. On the basis of this initial favorable response, we intend to convert several other training sessions to video tape. Initially, we will convert those sessions that involve training in the use of statistical packages and our courses on using the operating system on our principle academic computer.

# BI-DIRECTIONAL COMMUNICATIONS NETWORK DEVELOPMENT

Bruce A. Conroe

State University College, Potsdam, N.Y.

Potsdam College has been a leader in the preparation of computer programmers since the late 1960's when the major was first introduced under the Bachelor of Arts degree. It has been the philosophy of the College to merge theoretical learning with practical application as a way of recognizing the significance of the Liberal Arts. While the Computer and Information Science Department offers courses in scientific and business programming as well as information theory, other departments of the College utilize computer techniques and assignments to assist their instruction. In addition, the School of Professional Studies prepares teachers in uses of microcomputers, and upper level students in all disciplines make use of the computer in their learning. The College is also interested in having all students attain a level of computer literacy. It is vitally important that the campus computing environment be as advanced as possible. With this in mind, the College's Computer Advisory Committee put its energies to planning a bi-directional digital data communications network for the campus which could accommodate computer access instruments including an individual's own microprocessor as well as transmitting video signals.

Professors in such diverse areas as Music, History, Economics and Political Science have designed computer programs to aid their instruction. During the academic year 1979-80 the Computer Advisory Committee drew up a plan to move the College through three stages to meet the growing needs of academic computing. The first stage recommended immediate attention to the hardwiring of three terminals located strategically around the academic quadrangle to create small substations for academic computing in Chemistry, History, Political Science, Education and Computer Science. In addition, two large terminal rooms for general use were created.

The second stage called for the planning and conversion of the campus TV cable system into a two-way bussing network. Working with the Jerold Co.(original installers of the cable) the College's technicians in the Division of Media Arts and Technology wrote specifications for the equipment needed for this conversion. Computing Services staff were simultaneously working with the SYTEK Company to draw up plans for the equipment needed at the computer end of the network to make it functional. The cable was also drawn to the administration building making it part of the total network. Rooms are being established in various academic buildings and in the residence halls for a variety of pieces of equipment to form a cluster as well as having taps for attaching student's own microprocessors for access to the central Burrough 6810 Computer.

The third stage of the development will be finalization of 7 to 10 substations each with CRT's, printers, microprocessors and a minicomputer. Then, depending upon the level required, the student can utilize whatever computing he needs to do his work, including self-taught programs and other computer-assisted instruction. In addition to the campus network, students have access to three terminals wired to the IBM Computer at Clarkson College of Technology also located in Potsdam.

# LOW-COST PUPIL REPORTING SYSTEM FOR SCHOOLS

Claudia Devaux

LAD Computer Services

This report describes the rational behind the choice of hardware in the development of a computer system to be used by principals and teachers in the elementary school environment. We wish to point out a major pitfall that school districts may encounter when trying to keep the initial cost of computer systems down and emphasize the need for long term cost analysis when purchasing a computer system.

The system under development keeps track of student achievement test scores and produces class lists, diagnostic and prescriptive information, class profiles, and other reports. The task was to define the needs of potential users, identify the hardware required, and develop the software. A major goal was to keep the total system cost under $5,000.

The following hardware configuration was selected: Apple II-Plus computer with two floppy disc drives, an IDS-440 printer, a light pen ("Bright Pen" from Programma), and a videomonitor. The two floppy disc drives were required to permit random access and to ensure ease of backup. The light pen was an inexpensive alternative to keyboard data entry, providing input via menu driven screens. Besides being compatible with these peripherals and being viewed as supporting major educational software in the future, the Apple Plus-II was favored because of its ability to be interfaced to color television sets. Not counting the television set, and item already available in a significant number of schools, we were pleased that total hardware configuration was accessible at less than $3,835.

The software was developed according to specifications, and the system is intact. However, a school district may be disappointed with the system in terms of the number of floppy discs required to maintain the system, the overhead in backup procedures, and the fragile properties of the floppy discs. At this time we are re-evaluating hardware costs and looking into hard discs (E.G. Winchester technology), comparing costs of periods of three and five years.

# INTRODUCING COMPUTER BASED EDUCATION - PLATO

# AT THE UNIVERSITY OF ANTWERP

W. Dyck - University of Antwerp

F. Dams - University of Antwerp

Within the framework of the project "Improving the Effectiveness of Undergraduate Education" the PLATO - CBE system was introduced in 1980 at the University of Antwerp. It is important to know that there are no real restrictive regulations for admission to a Belgian university. This results in a very heterogeneous group of freshmen. These students are confronted with a teaching-learning model that has been compared to a neo-Spartan contest. The outcome is a 50% failure rate. Our search through the educational literature led to the conclusion that efforts to improve effectiveness within the so-called "dominant prediction-selection paradigm" were never strikingly successful. So we planned to reorient instruction in the direction of a mastery learning strategy. As one might expect, the social environment turned out to be extremely sceptical about the probability of success in our attempts to improve outcomes. Faculty members were not prepared to commit themselves to a full blown mastery learning project and the students still enjoy the indefeasible right to work for their study or not. Therefore, we imposed a pragmatical and carefully planned innovation strategy, directed towards a gradual increase of impact on the curriculum. The PLATO-system was one important element in this strategy. First of all, the electronic library of PLATO was evaluated and different lessons were selected. It was announced that freshmen could make use of one of the five terminals some months before the beginning of the academic year. The reason for this is obvious: the overwhelming importance of the cognitive entry level. Research shows that entry tests are very predictive. Thus the first efforts were concentrated on lowering the correlation between entry characteristics and examination results. Our specific contribution was the creation of a programmed test instruction that fulfilled the basic math, physics and chemistry requirements for freshmen medical and veterinary school, dentistry, pharmacy, chemistry and biology curricula. For every distractor of the 140 5-options multiple choice items a possible diagnosis was written and correctives were given. The student can ask at any time for the theoretical background; when he makes a mistake or calls for the theory, additional exercices must be done before he or she receives the next item. This lesson is programmed in "Tutor" and we use it as an instrument of formative self-evaluation. Also in this lesson several references are made to the published courseware of PLATO. This pre-university phase of our research is followed by applying the computer for remedial and complementary instruction during the academic year. For each academic course booklets of formative self-evaluation tests are given to the students. A total of approximately 1000 multiple-choice items cover e.g. the freshmen medical curriculum. Representative samples of these items were entered via the PLM-driver to control the mastery level of the self-evaluation. Here also the PLATO library functions as a medium for supplementary instruction. Overall effectiveness, test results and responses to questionnaires will be reported.

# COMPUTER-ASSISTED INSTRUCTION AND COMPUTER-MANAGED INSTRUCTION PROJECT

Margaret "Kit" Ellis

Seattle Central Community College

In the past three years Seattle Central Community College has established a successful program of computer education to help meet the institution's long range goal of providing improved educational services to students. The program, funded by the Department of Education Title III Advanced Institutional Development Program, addresses four main areas: computer-assisted instruction, computer-managed instruction, faculty training, and computer courses.

Establishing computer-assisted instruction (CAI) involved obtaining hardware and software to support use by any disciplines which were seriously interested. Most CAI is now supported on 16 Apple II Plus microcomputers. Most programs were purchased; some were written locally.

Computer-managed instruction (CMI) was started to assist instructors in keeping useful records for the large self-paced basic skills labs and to enable early warning and intervention for students experiencing difficulty. CMI is now used in the mathematics, reading, writing and English as a second language laboratories. The systems are supported on two Alpha Micro computers, each with a ten megabyte disk drive.

Faculty training has been an integral part of the whole process of introducing computers into instruction on a college-wide basis. More than 80 faculty have taken at least one 5 week/10 hour introductory course on computer use in education. Most of those have taken at least one additional course that introduced programming in BASIC and/or PILOT. All of the locally produced courseware has been produced by these faculty members. Every faculty member has been exposed to at least one orientation session on instructional computing.

Several new computer-related courses have been added to the curriculum. They include basic computer literacy, languages and computer science. They are distinct from the vocational data processing courses that have been available at the College.

The effort has been successful. Retention rates have increased through the use of CAI and CMI. Better record keeping has enabled early warning and intervention for the basic skills students. Through CAI and computer classes the College has provided our students with valuable learning experiences that would not have been possible without the use of computers. The level of computer awareness among all constituencies; faculty, staff, and students; has risen dramatically. The College has been able to provide improved instructional services to students through the use of computers.

# The NICE Project -- A Case Study in Innovation

Annalee Elman - American Institutes for

Research

## ABSTRACT

This article is a progress report on the establishment of a network to foster communication between educators and industry on plans and cooperative activities to promote computer literacy.

NICE goals and objectives are identified, and activities and achievements to date described. For prospective innovators, ingredients for a successful campaign are enumerated.

END

The concept of a Network for Industry, Computers, and Education was born out of a perception that

- education was being uncommonly slow in adopting the new microprocessor-based technologies in the pre-college curriculum
- industry had and has a vested interest in seeing computers used widely in schools.

From these observations, it seemed evident that a partnership between industry and education around a common goal of increasing the penetration of computers into school curricula might be useful.

To test this belief, I wrote a letter to the editor of Computer World, the leading weekly publication for computer professionals, inviting dialogue and discussion on what schools ought to be doing to prepare our children for the future. I got some interesting replies. A systems analyst for AT and T Longlines wrote to tell about his experiences teaching BASIC to fifth-grade students in San Francisco. His teaching activities are sponsored by CAPS (Corporate Action in Public Schools) of which AT and T is an affiliate. A Management Information Systems chief in a state government department expressed his concern over the distance growing between computer-literate students and other students. A New York City systems designer pointed out the power of the computer as a tool to help people organize themselves. Also from New York, a senior systems analyst decried the poor communications between data processing and the user, and between data processing management and technical people. He blamed faulty communication between non-technical and technical people on computer illiteracy, and provided some illuminating examples of the consequences of such illiteracy on getting a job done.

One of the correspondents, a programmer analyst in Florida, asked:

1. Do you realize how far apart academia and the real world are?
2. How can we ('the real world') impress upon those in education of the 'practicality' gap?
3. How can we assist them in upgrading and/or establishing 'real world' courses of instruction?

An Illinois businessman averred that "the salvation of public education... lies in the commitment of ourselves, business people, the communities supposedly being served by public education." He went on to say that "...laying blame and remaining uninvolved will have far more deleterious effects than simply watching our public education institutions decay."

My next step was to begin a compaign to bring financial support to the aid of the network. This has turned into a full-time effort, and has led me (and a recently-acquired assistant) along some very promising paths. In the process, we've contacted people in the computer and publishing industries, in various departments of the federal government, in county offices of education, and on the staff of several foundations. We've met with some rebuffs and rejections; but most people we talk to are supportive and helpful. We've visited classrooms where computers are being used; had hands-on experience with every microcomputer offered for classroom or personal use, had demonstrations of software and courseware. We've also turned up several new business ideas and interested a number of companies in them. These leads have come as "spin-offs" from the original concept of the network.

What have we accomplished?

1. AIR staff are now computer-aware. Thanks to a series of lunch-time demonstrations, of hardware and courseware, we've been exposed to the Commodore PET, the TRS-80, the ATARI 800, the TI 99/4A, and the Apple II. We've seen courseware by SRA, McGraw-Hill, and Scott Foresman, demonstrations of LOGO, PILOT, and BASIC, and management and financial software for micros. The hardware and software vendors were all very cooperative in setting up and conducting these demonstrations, and the in-house response was excellent - 30 to 40 people attended each of the six sessions.
2. We have an extensive mail and telephone contact list, consisting of people in education (instructors in junior colleges and colleges, elementary and secondary school teachers, and district, county, and SEA personnel) senior corporate officials in computer hardware and publishing companies, technical writers, information services, computer professionals in corporations small and large, and state and county data processing departments. We keep in touch with our contacts, share information with them and seek advice and opinions from them. They are the network.
3. We have started a newsletter to disseminate information of interest to our network members. Because of our position in the research community, and our extensive and growing library, we are able to provide information not generally available to our networkers.
4. We have polled our members on what directions they would like to see the network take.
5. We have written and submitted prospectuses and proposals both for the network and for several other related projects which we discovered as a consequence of our contacts..
6. We are working on developing several new concepts.

In the process, we have generated a lot of interest, excitement, and involvement, and far more work than we can handle with our limited staff and resources.

How have I managed to get this far? Several factors have been instrumental in the degree of success achieved so far. I pass them on as important ingredients to getting any new computer effort off the ground:

They are:

- Do your homework. Know your facts, and keep informed on recent happenings in the field. Know who's doing what, and what the trends are. This can be accomplished by reading. I've made an investment in personal subscriptions to over a dozen magazines and journals. My organization has subscribed to a half-dozen others. Most of these can be obtained in the reading room of a large public library or on newsstands. Get your library to subscribe, if you don't want or can't afford to do so yourself.
- Be organized. Clip and file any thing that strikes your fancy. Keep records of every contact made, with dates, addresses, and telephone numbers.
- Communicate. Keep your colleagues and management informed of your activities on a regular basis. Do this in writing. Seek out and accept invitations to present your activities, plans, and problems to groups within your organization. Ask your friends to pass on to you anything they come across that they think is relevant to your interests.
- Be enthusiastic. Not hard to do if you're informed.
- Learn to program. Many community colleges offer courses in beginning programming. Or, you may find workshops offered at local computer stores. Place a few phone calls to nearby colleges. Call your school district offices. Ask people who already program for their advice.
- Talk about what you're doing and thinking of doing with everyone you contact. Someone you speak to may be able to offer useful suggestions. Plan to make follow-up contacts to keep people informed of your progress, and thank them for their help. You never know when someone you've written to or spoken to may give you just the referral, piece of information, suggestion, or encouragement you need.

- Don't give up. There's a psychological barrier to be overcome which you will encounter sooner or later. It comes when you've encountered yet another dead end, are tired or otherwise discouraged. Acknowledge the doubts about the viability of your attempts to innovate as the "devils" that come with territory. Dismiss them without a second thought.
- Set your sights high. While what we've accomplished thus far falls far short of our initial objective, I'm persuaded that we would not have gotten as far as we have if we had started with less ambitious goals. With continued effort and some good luck, I expect to be able to be back next year with another chapter written in our success story.

# GRAPHICAL EXPLORATIONS (PROJECT REPORT)

Robert Evans - Cuyamaca College

A series of modules (programs) have been developed which allow the student to easily explore the relationships between the algebraic and graphic forms of all the functions and relations usually considered in precalculus mathematics including:

1. Cartesian coordinate system
2. Linear equations
3. Polynomials
4. Rational functions
5. Conic Sections
6. Logarithmic functions
7. Exponential functions
8. Polar coordinate system
9. Trig functions

Each module has three operational modes:

EXPLORATION - Allows the student complete control of all functional parameters through easily operated menus or commands. For example, in the linear module the student has control of the slope and intercepts. Once the parameters are selected and values provided, an appropriately scaled high resolution graph is produced. Graphs can be overlaid to study the effect of a particular parameter. No computer or programming knowledge is assumed or required.

GAME - Automatically sets the parameters within the function being studied. An appropriately scaled graph is presented to the student. The student is then presented a question concerning the graph and the value of one of the functional parameters. For example, if a straight line were displayed, one possible question would be: What is the slope? Responses are timed and a score is assigned based on accuracy and speed. A fixed number of problems constitute a game.

QUIZ - Similar to the game mode. However, in this mode the student is not timed and student responses are logged for later scoring and analysis by another program available only to the faculty member.

The use of a "discovery" approach has been received enthusiastically by the students. Student insight into the relationships between functional parameters and graphical form appears more complete. Finally, the use of the computer and its graphic capability has reduced the time for students to develop the necessary insights and proficiencies related to translating from one form of a function to another.

# CREDIT AGRICOLE

J. Favry

IFCAM, PARIS

At the beginning of 1982, 130 Micro Plato stations were installed launching the introduction of CBE in our bank "Le CREDIT AGRICOLE MUTUEL" (C.A.M.).

Our bank employs 67,000 people is one of the leading banks in the world. The decision to proceed with CBE was reached through a 3 year process conducted by IFCAM (training institute of the CAM). Studies, research and experiments are described below.

## 1. The Studies

1.1 The study of opportunity. After studying the opportunities, we have come to the conclusion that CBE was needed in our firm. This was primarily an education project. The choice of equipment is only a consequence of this project.

1.2 Study of feasibility. This study was to investigate whether both the education site and the educational policy of C.A.M. were the right ones for C.B.E. The answer was positive.

## 2. Research

2.1 Following the results of the above study of opportunity, a pedagogical project was defined: in-house self-training assisted by a computer so that the employees can be trained thanks to various teaching techniques.

2.2 Through this project we were able to determine the pedagogical quality necessary for C.B.E. lessons in C.A.M.

2.2.1 The general conception of a lesson is directly inspired from teaching by objectives.

2.2.2 Several pedagogical criteria have been chosen for reference. Three can be measured quantatively (ratio): interactivity, individualization and initiative. A special training is given to the teachers to teach them about the specific methods of C.B.E.

2.2.3 The pedagogical demands have resulted in establishing criteria for the choice of a system for this project = Micro Plato of Control Data has been chosen.

## 3. Experiment

The two experiments were conducted in a real setting within the frame of the project.

3.1 The first experiment analysed the efficiency of C.B.E., measuring the quantity of acquired knowledge. It took place according to precise rules based on statistical experiments.

3.2 The second experiment enabled us to discover the attributes of C.B.E.: motivations and restraints of learners. On the whole we can affirm that C.B.E. is motivating and attracting.

## 4. Development

The forecast has been established for three years. In 1985, 300 Micro Plato stations should be installed in the bank agencies. At the same time, the total number of courses will amount to a hundred lessons which will be used by almost 2,000 employees. If C.B.E. is a success these figures could be tripled in the next project. At the same time, other pedagogical projects for the use of C.B.E. will be studies.

# HAND HELD COMPUTER (HHC) TRAINING APPLICATIONS

Frank E. Giunti, Donald A. Kimberlin

TDI and Ordnance School

## PROBLEM:

The increasing level of complexity of military systems and operations has resulted in dramatic increases in technical documentation and decision making requirements. The advent of micro circuitry technology with the resultant development of the hand-held size computer offers a means of bringing significant computer power to the job, no matter where the job is. The new asset must be examined to determine its feasibility job-site use and to find its appropriate training application.

## OBJECTIVE:

Determine if the HHC can be utilized as a viable automated job aid for training and job site applications by taking advantage of its computational, electronic storage and decision making capabilities.

## DISCUSSION:

Phase I of the HHC project was divided into two tasks. Task I of this project surveyed in detail the available HHC and compared their features with regard to possible uses in training and job site applications. All HHC marketed or soon to be announced were considered. With this survey information available to them, the research investigators then proceeded with Task II which involved the survey of selected Army courses taught at the Ordnance School, Aberdeen Proving Ground, MD and identified specific training tasks that could be employed to demonstrate the feasibility of using HHC for training and job site applications. The general areas of application that were considered included maintenance and troubleshooting, maintenance management, quality control procedures, basic electronics and logic training, problem solving for circuit analysis, logic mechanization problems, training management for testing and for diagnosing student learning problems. Based on the survey results of Task I and II and the positive recommendations made by the research evaluators and accepted by TDI and the Ordnance School, Phase II of the project will begin. This will involve a test and evaluation of the selected HHC and the appropriate courses identified in Phase I in an effort to determine if training and job site applications are effectively enhanced by the use of the HHC.

# COMPUTER BASED INSTRUCTION (CBI) FOR THE TOW FIELD

Frank Giunti, Donald A. Kimberlin

TDI & Missile and Munitions School

## PROBLEM:

The TOW Missile Field Test Set is a limited issue item and as such, is in short supply in the training base. Additionally, if certain test sequences are not followed, there is an increased possibility of damage to the test set and extensive maintenance down time. These factors limit hands-on training, reducing training effectiveness and the quality of the schools' graduates.

## OBJECTIVE:

To evaluate the training effectiveness of employing Computer Assisted Instruction (CAI) and two dimensional simulation courseware to train using the Control Data Corporation (CDC) PLATO System and Micro PLATO terminal.

## DISCUSSION:

Approximately 20 instructional hours of maintenance will be provided using CAI techniques and two dimensional simulations. The students will be permitted to make tests and execute procedures not now permitted for safety and maintenance reasons. Student input will be through the use of the touch panel and the keyboard. The TOW students will be using the Micro PLATO system instead of the PLATO terminals tied to a time shared computer system. The Micro PLATO offers quicker student response time, more portability, and freedom from the problem of a time shared system.

## RESULTS:

Positive results of this test will provide a basis for using CBI as an equipment extender in those training programs where expensive equipment is limited, or even non-existant for training, or hazardous to use with inexperienced personnel. Projected completion date for this project is scheduled for December 1982.

# DISTRIBUTED INSTRUCTIONAL SYSTEM

Frank E. Giunti, Donald A. Kimberlin

TDI & US Army Air Defense School

## PROBLEM:

The improved Hawk (I-HAWK) Fire Control Mechanic Course in the US Army Air Defense School, Ft. Bliss, TX includes hands-on training and troubleshooting of the High Power Radar Set from the front panel of the Battery Control Central. There is a limited number of systems available for training and those that are used suffer from excessive down time and high maintenance costs due to extensive student interaction during training. This limits the necessary "hands-on" training that can be provided to the students, thereby reducing training effectiveness.

## OBJECTIVE:

Determine if an interactive microcomputer/videodisc based DIS could be developed to provide the essential "hands-on" training in a training and cost effective manner.

## DISCUSSION:

WICAT, Inc., under contract to the US Army, is developing a DIS consisting of one store station and four work stations. The store station will maintain student records and serve as a courseware library. Each work station will have local storage on a floppy disc and will communicate with the store station as required to pass records up and down and obtain course material. Two types of training programs will be used with the system, a Computer Assisted Instruction (CAI) program to teach the seven system checks for the I-HAWK system, and a two-dimensional (2D) simulation will be used to teach the system integration check using "hands-on" training for the high Power Radar Set. It is during this training session that the student will be learning troubleshooting techniques, normally taught on the actual equipment.

## RESULTS:

## RESULTS:

Positive results on the outcome of this project will become the basis for expanding the application to other areas of the I-HAWK System and will offer the potential for similar applications in active air defense units where hands-on training is limited because of the operational necessity of having the air defense in a constant ready posture. Experiences gained from this test will also have TRADOC-wide applications to similar Air Defense, Missile, and Field Artillery Systems to include the initial 2D simulation model development. Estimated completion date for this project is scheduled for June 1982.

# VIDEODISC INTERPERSONAL STRESS TRAINING ASSESSMENT

Frank E. Giunti, Donald A. Kimberlin

TDI & US Army Infantry School

## PROBLEM:

The Basic Infantry Officers Course includes training in techniques of counseling subordinates. The current training program is conducted by lecture and role playing. The role playing element is highly labor intensive because each group of four to five students is guided by a faculty member; the training quality is highly dependent on group interaction and the training is difficult to standardize.

## OBJECTIVE:

Examine the viability of using a microcomputer videodisc (MC/VD) system to provide vicarious experiences of leader-subordinate interactions to train Junior Officers and Senior NCO in interpersonal leadership and counseling skills required to deal with typical stress situations they will normally encounter on the job.

## DISCUSSION:

The Army Research Institute, Ft. Benning, GA, is investigating the viability of using an interactive microprocessor/videodisc training delivery system to teach students in the Infantry Officer Basic Course, Infantry School, Ft. Benning, how to deal with some of the most common interpersonal stress situations they will encounter as junior officers. The following scenarios have been developed for evaluation. Verbal abuse, taking charge, NCO responsibilities, performance counseling on NCO, insubordination, and crisis referrals. The scenarios set the situation for the student. As the student selects his response to a stimulus, the probable result will be displayed. The student is permitted to check the results of each response or select only the areas he believes are most correct. In any case, there is remediation for each improper student response. There can be a quasi dialogue developed if the student selects only proper responses. An example is the scenario on verbal abuse. The scene is set by the lieutenant observing one of his squad leaders berating a soldier in an abusive manner. One option, an improper one, the lieutenant corrects the sergeant on the spot before his squad. The scene reflects the squads amusement and the sergeant's immediate hostility. A selection of the proper option has the lieutenant speak to the sergeant aside from his squad. In each case the scene can be viewed from the lieutenant's perspective.

## RESULTS:

Positive results in the evaluation will become: the basis for full implementation of the program in curriculum of the Infantry Officer Basic Course; training package for other Army schools that provide Officer Basic Training; and can provide a basis for further test and evaluations in such areas as the unique situations following the new female officer, and soldier's perspective of the command relationships and responsibilities. Estimated completion time for this project is scheduled for January 1983.

# EQUIPMENT INDEPENDENT TRAINING SYSTEM

Frank E. Giunti, Donald A. Kimberlin

TDI & US Army Signal School

## PROBLEM:

Training Satellite Ground Station Repair personnel using actual equipment pose several significiant problems. These include: degraded training when a system component is "down" for maintenance or repair, high on-going maintenance costs, and student queuing when waiting for "hands-on" training, reducing training efficiency. To obtain additional satellite stations at $14 million each is cost prohibitive. Equipment specific simulators would cost nearly as much to develop and obtain as the actual system and the Army would continue to be plaqued with maintenance and equipment obsolescence problems.

## OBJECTIVE:

Determine if microcomputer/videodisc (MC/VD) technology can provide a cost effective interactive two dimension (2D), equipment independent training system that would be training efficient and effective.

## DISCUSSION:

The US Army Signal School, Fort Gordon, developed a 2D simulation of a major component of the satellite system, the AN/FTC-98 multiplexer. The purpose was to provide "hands-on" training in the procedures for power on/off and alignment, and system self-test operations. Student interface with these procedures and operational checks on the EIT System was primarily through the use of a light pen. Using appropriate TV production features as "cut to" from full front panel displays to specific quadrants and split screen, also enhanced student interaction. Prompts and remediations were also built into the lesson. The simulation averaged 13 minutes per student to achieve mastery of practice activity. Two groups of 30 students each were used for the test. The control group was provided "hands-on" training using the MC/VDEITS. Both groups took the same written and performance tests. The experimental and control groups were given the performance test on the actual equipment. This was the first "hands-on" exposure on the multiplexer for the experimental group. The evaluation took place under normal school operating conditions, consequently all variables could not be controlled.

## RESULTS:

The two groups were evenly matched. There were no significant differences between groups in time to complete the training, time to complete the tests, or in achievement on the written performance tests. Students and faculty acceptance was high. A key factor is the potential for significant cost benefits in reduced equipment maintenance and acquisition. These cost reductions make EITS an attractive training alternative in place of the use of actual high cost equipment.

# APPLICATIONS OF MODERN TECHNOLOGY TO ARMY TRAINING

Frank E. Giunti, Donald A. Kimberlin

IDD Training Developments Institute

## MISSION:

The IDD is responsible for assisting and supporting Army Schools, Training Centers, and Reserve Components in their training mission. It is also responsible for institutionalizing the Systems Approach to Training as applied to course development and delivery and test evaluation of modern and emerging technology to training, training management and technical documentation.

## PROBLEM:

The Army faces a great challenge to provide a well trained modern force that must be mobilized at a moments notice. This requires initial and sustainment training in support of job proficiency for the current weapons systems and some 300 new systems scheduled to be fielded in the next 10 years. Since these systems are sophisticated, complex, and incorporate the latest technologies, so too must the Army Training Systems employ the latest technology to effectively train individual and collective critical tasks required of the soldier in today's modern Army.

## CURRENT APPLICATIONS:

Current IDD is engaged in 18 various projects with the schools and training centers involving a variety of technologies. These range from the investigation of individualized typing training consoles to teach and maintain the typing skills of Administrative Specialists and Teletype Operators, to low cost two dimension (2D) simulation delivery systems, to the Training of Missile maintenance personnel in the TOW Field Test Set using CAI and 2D simulation. A select number of these projects will be presented during the Seven Minute Project Report sessions the next two days.

## FUTURE APPLICATIONS:

In a continuing effort to apply emerging technology to training and technical documentation and determine their cost and training effectiveness, TDI working closely with the Army Communicative Technology Office, will continue to test, evaluate, and implement the proven technologies. Examples of the technologies that are currently in the Research and Development (R&D) stages and are being considered are: flat screen television, digital memory for videodisc, voice technology to include speech recognition and synthesized speech, surrogate travel, and the interim and long range development of an electronic development, distribution and delivery systems for training and technical documentation. It is anticipated that technologies such as these will assist in providing the nation with a well trained, highly skilled Army to meet challenges of the next decade.

# PROGRESS TO DATE ON THE IMPLEMENTATION OF THE INTEGRATED CYBERNETIC MUSIC SYSTEM (ICMS)

Sherwin Gooch - Florida State University

At the recent ADCIS convention in Virginia, Steven R. Newcomb and Sherwin Gooch of the Center for Music Research (CMR) announced the CMR's intention to begin a laboratory to build the Integrated Cybernetic Music System (ICMS). The ICMS can be viewed as a collection of sub-systems (both hardware and software), each accomplishing a specific task for the musician. The foundation of the ICMS is to be a research database capable of containing all published Western-style music in a score-reproduceable format. The musician may interact with the database in many ways, which are most easily classified as input and output (I/O) of sound, I/O of music scores, and I/O of textual information. Prototypes of most of the components required for the ICMS have been completed. An inventory of the work completed, and the work remaining will be presented.

# THE UNIVERSITY OF TEXAS AT AUSTIN SCHOOL OF NURSING

# COMPUTER ASSISTED INSTRUCTION PROJECT

Susan J. Grobe, John R. Tarp

This report will highlight the computer based Instructional Module Authoring System being developed at the U.T. School of Nursing under a grant from the Division of Nursing Special Projects Grant Section of the Department of Health and Human Services.

The purpose of the project is to develop, field test, and evaluate a CAI module authoring system. The system is designed for nursing faculty's use in preparing CAI modules structured around five steps of the nursing process: assessment, diagnosis, planning, intervention, and evaluation. The system will allow faculty who are unfamiliar with both instructional design and computer programming to plan and develop computer assisted instructional modules in nursing.

Participants will be briefed on Bloom's model of instruction, which underlies the system. Examples from the project will be used to illustrate how broad areas of nursing content can be broken down into discrete planning units.

Handouts will be available that summarize the authoring process from an instructional perspective, and outline the use of this process to make development of instructional programs easier, faster, and more instructionally sound.

# MICROCOMPUTERS IN MUSIC-THEORY CAI:

# A FIRST RETROSPECTIVE

Dorothy Gross

University of Minnesota

In a brief five years, microcomputers have had an amazingly strong impact on academic computing. Despite questions raised about the usefulness of the rudimentary programs that might result from smaller memory size, the affordability of microcomputers has made computing possible for more people than ever before. Music theory had been a prime area for many computer applications to CAI since theory lent itself to logical formulations and drill-and-practice lessons, so it is hardly surprising that a large part of the musical courseware in CAI for the smaller machines has concerned theoretical subjects. But in our haste to design programs for the new machines, perhaps it is time to pause momentarily and consider those original doubts about microcomputers. How has the significant loss of memory space affected the quality of instruction?

While many types of small computers have been used, the one used most widely for music applications appears to be the Apple II, probably due to its built-in tone generator and a store of commercially available music software. This paper will consider some programs for the Apple II that have been in use for some time. Specifically, I will discuss lessons by Linda Borry for music fundamentals, Bruce Benward for harmony, and Charles Boody for ear training. All three sets of lessons are student-tested; hence they should be free of technical problems and present a clear picture of the designers' musical and pedagogical intentions. Lessons will be evaluated according to instructional design, student results, student use, and instructor opinion. Following this evaluation, the paper will conclude with a discussion of microcomputers in music-theory CAI in general, drawing from other courseware as well as those described in detail above.

# A REAL-TIME FUNDAMENTAL PITCH DETECTOR FOR USE IN COMPUTER-BASED MUSIC LEARNING SYSTEMS

Fred T. Hofstetter

University of Delaware

Since 1980, the University of Delaware has been working with the Kay Elemetrics Corporation on an interface between Kay's Visi-Pitch and the IST family of programmable PLATO terminals. The Visi-Pitch is a real-time fundamental frequency and relative intensity analyzer that operates in four frequency ranges with analytical circuitry that is specially designed for analyzing the human voice. The four frequency ranges correspond to soprano, alto, tenor, and bass voices.

When students sing into a microphone connected to the Visi-Pitch, information about fundamental frequency and relative intensity is immediately displayed on a Tektronix 5111 storage oscilloscope that is built right into the Visi-Pitch. The split screen capability of the scope allows the user to erase and store information independently on the upper or lower screen.

Through an interface specified by Bob Stradling of the University of Delaware's Office of Computer-Based Instruction (OCBI) and added to the Visi-Pitch under a special contract with the OCBI, the Visi-Pitch connects to the parallel port of the IST PLATO terminal and passes pitch period information to the terminal at a rate of 1/Fo samples per second. A micro-TUTOR program written by Adam Baum, a student programmer in OCBI, computes a moving average of the pitch period information sent by the Visi-Pitch and makes the resulting pitch available for processing by the GUIDO aural-skills lessons programmed by Bill Lynch, a programmer analyst in OCBI.

Adam Baum has converted the GUIDO Intervals program so that it works with the Visi-Pitch, and a tolerance variable has been added to the table that drives the Intervals program so that the pitch accuracy required of the students can be adjusted as the GUIDO units increase in levels of difficulty. When students sing an interval, the GUIDO program tells them whether they are correct, sharp, flat, or on the wrong pitch altogether for each note in the interval. When they sing the wrong note, GUIDO tells them which note they sang by showing it on a piano keyboard and displaying pointers that tell the students whether they are too high or too low. When students sing the right note but are flat or sharp, GUIDO displays a flat or a sharp of the correct pitch on the keyboard.

The GUIDO melody program is also being converted to run with the Visi-Pitch.

# TOOLS - IDENTIFICATION AND USE IN INDUSTRIAL ARTS EDUCATION

## A PLATO PROJECT

J. F. D. Ilott - University of Alberta

The Department of Industrial and Vocational Education at the University of Alberta prepares industrial arts and vocational teachers for junior and senior high schools. Mastery of basic knowledge about tools and their use in a metal shop is the focus of this PLATO course, which has been incorporated into a beginning level teacher education course.

The project has three groups of objectives. First, there are what might be called project objectives. These include a check on the feasibility and effectiveness of this kind of instruction through the medium of PLATO. Originally the course had been developed on the IBM 1500 CAI system; this project includes a revision of the earlier course. Second, objectives for the course instructor include freeing him from teaching this low level cognitive material and allowing him to devote more time to teaching psychomotor skills in the workshop. A further objective is to relieve the instructor of the less interesting parts of a beginning course. Third, from a student viewpoint, the objectives are to provide systematic, thorough instruction that may be accessed in an unscheduled manner, that may be reviewed at any time, and that can serve as a review for the second level course taken later in the student program.

The equipment used in the project includes standard PLATO terminals (IST-II) linked to a ReVox microfische projector. These units, which also include a tape player, with both the tape player and the projector controlled by the TUTOR program, are used by education and medicine students at the University of Alberta. The terminals are, or course, linked to a Control Data Computer.

Program features include a driver program developed for the project. The course is menu-driven in the sense that students select a particular lesson, but once a lesson has been selected the driver program takes over. Having selected a lesson, students first complete a pretest; then, if necessary, are taken through the instruction and complete a post test. Branching and looping, depending on student answer responses, are a major feature of the program.

# EVALUATION OF A SPATIAL DATA MANAGEMENT SYSTEM FOR BASIC SKILLS EDUCATION

Greg P. Kearsley

Human Resources Research Organization

"Evaluation of a Spatial Data Management System for Basic Skills Education" is a forty-two month phased project to develop and evaluate applications of an advanced multimedia technology for basic skills Management System (SDMS), which incorporates videodisc, micro-processor, and microfiche. The project consists of two phases.

The first phase involves a comprehensive research and development program in the area of spatial orientation skills and learning strategies. Applications of SDMS features include the use of the SDMS surrogate travel capability to present learning material in orienting and navigation, and the use of the SDMS interactive movie capability for teaching test anxiety-reduction and study-skill learning strategies.

In the second phase of the project, the focus will be on the attainment of basic skills--reading, auding, and computation--specifically related to a cluster of occupational specialties. The multimedia capabilities of the SDMS will be used to present realistic job contexts, and to sequence visuals, sound, and prose for functional literacy training based on cognitive development theories. The variety of media and techniques available in the SDMS will permit the development of enjoyable and motivating presentations using games and humor.

The materials developed will be used to explore the learning processes underlying the acquisition of these skills and knowledges. More specifically, the following questions will be studied:

What information organization is most conducive for learning spatial orientation skills for different types of learners?

What improvements in test performance can be attributed to learning strategies designed to reduce test anxiety and improve study skills, and to the attainment of test-taking skills?

Which navigational learning strategies are effective for what types of learners?

Can the replication of a cognitive skill development model in a functional literacy program enhance the acquisition of baseline reading skills by soldiers in BSEP?

Can the motivating aspects of the SDMS (learning control over information, "entertaining" multimedia presentation techniques) be capitalized upon to enhance learning of MOS baseline computation skills?

# CROSS LINGUAL MATHEMATICS INFORMATION SYSTEM

## (PROJECT REPORT)

Jim Kelly - Evanston, Illinois

A simple but useful data base has been developed with a random sequential file consisting of 1000 records, each containing the English, Spanish, and French equivalents of commonly used conceptual, operational, and notational elementary and secondary school mathematics terms. The information system will serve the classroom teacher, the curriculum designer, and the mathematics education community in general.

The project is in three phases. The first phase consists of a program which allows the user to do direct searches or category searches for the language equivalent of any of the 1000 terms. The software was written for an Apple II (48K) with two disk drives and monitor. The second phase is the accumulation of content information for a basic information retrieval system. The content of 152 elementary and secondary school books from five countries has been compiled into a 24,000 record file. Each of the 19 byte records shows, through special coding, where and how the mathematics terms are used in the books. The software developed accesses the terms file of phase one, selected portions of the content file, and a small 125 byte bibliographic information file to give the user cross lingual information either by term, book or grade level.

Work on phase three is just beginning. It will consist of multiple language scope and sequence creating, comparative methods on examining curricula, and curriculum designing simulations. A printer and hard disk will be added to the Apple II.

# FIELD TEST OF A MICROCOMPUTER BASED SYSTEM FOR COMPUTER ASSISTED INSTRUCTION IN MUSIC

Wolf Kuhn - Stanford University

P. Lorton, Jr. - University of San Francisco

At the ADCIS 1981 Conference, we reported on a system for computer assisted instruction in basic musicianship which we were developing for use on an Apple II microcomputer with an alpha-Syntauri keyboard. During the past year, that system has been field tested at a variety of sites and we have learned some practical lessons about providing this sort of material to users in the field. The purpose of this report is to summarize the feedback we have received and highlight the lessons we have learned.

The CAI in Basic Musicianship program is modeled on the work done during the last ten years for the Music Department, Stanford University. Using an Apple II microcomputer, a Mountain Computer Music System synthesizer and an alphaSyntauri keyboard all provided by the Foundation for the Advancement of Computer-based Education, the program provides drill and practice in some of the areas fundamental to basic music theory - scales, intervals and triads. (A more complete description of the system is available in Kuhn & Lorton 1981).

During the 1981-82 school year, several institutions have tested this material. In general, their feedback has been quite positive for the concept and full of suggestions for improvement and enhancement.

Attempting to produce a CAI system which would be available quite widely and which would use fairly sophisticated additions to the Apple II has provided several lessons which will be quite useful for further development of the system. These lessons concern such things as the amount of on-line support required for a "stand-alone" system, the problems associated with making a microcomputer program absolutely flawless, and the desirability of built-in flexibility.

The experience of this year has provided a valuable balance to the academic, research oriented view with which we began this project.

Reference: Kuhn, W. & P. Lorton, Jr. "A Microcomputer-based System for Basic Musicianship". Proceedings - Annual Conference of the Association for the Development of Computer-based Instructional Systems, Atlanta, GA., 1981. Pp 267-270.

# MATHEMATICS IN BIOLOGY: COMPUTER-CONTROLLED VIDEODISC MATERIALS FOR WOMEN AND MINORITIES

James M. Laffey

University of Washington

The general purpose of the "Mathematics in Biology" project is to produce an instructional package which makes use of microcomputer and videodisc technology to teach basic, secondary-level mathematics. The target groups for this instructional package are women and minorities who have a career interest in the life and health sciences, but whose current level of competence in mathematics is inadequate to undertake the post-secondary education required for such a career. We seek to meet the special needs of our target groups by teaching mathematics within the context of their career interest and by utilizing technology to increase the breadth and quality of interaction between the student and the curriculum.

Our project received a DISE award from the National Science Foundation and work started in October, 1981. We are using an Apple II Plus 48K microcomputer and an industrial videodisc player from Discovision Associates. We intend to produce four units of instruction: "fractions", "ratios and proportions", "measurement and dimensions", and "functions and graphs". The student will interact with this instruction via software written in UCSD Pascal, an optical videodisc, and a companion workbook.

We propose a "project report" which would discuss the progress of our work. Although our project will not be complete until Fall, 1982, we expect to have pressed a preliminary disc and to have some evaluation information by the end of May, 1982. We would be pleased to describe the stages of our instructional development work, the problems we have encountered, and our vision of the worth and potential application of microcomputer/videodisc instruction.

# THE PLATO CALCULUS COURSE (PROJECT REPORT)

Max D. Larsen

University of Nebraska-Lincoln

Five universities have joined together to produce a computer-based instruction course in Calculus. The five universities and individuals involved are the following:

- John Bergman, Department of Mathematical Sciences, University of Delaware
- Wally Etterbeek, Department of Mathematics, California State U.-Sacramento
- David Lomen, Department of Mathematics, University of Arizona
- Charles McCarthy, Department of Mathematics, University of Minnesota
- Max Larsen, Department of Mathematics & Statistics, University of Nebraska-Lincoln

These five individuals have formed the Editorial Review Board appointed by Control Data Corporation for the purpose of producing a full calculus course delivered using PLATO.

The Editorial Review Board has written instructional objectives and lesson content descriptions for the first semester course. Preparation of programmable ready material and the actual programming will be performed by the Office of Computer-based Instruction at the University of Delaware. The lessons produced will be reviewed carefully by members of the Editorial Review Board and field tested at their respective universities. The lessons for the first semester course should be available for distribution by Control Data Corporation by fall, 1983. Preparation of second and third semester courses will follow.

Each of the courses is modular in structure and can be used either as a stand-alone full course or as a supplement to an existing course. The student study guide will be keyed to five major textbooks. Testing will be done on a central PLATO system and credit awarded by the institution subscribing to the course. Lessons and drill will be offered on micro-PLATO stations using floppy disks.

In addition to the availability of the stand-alone course, this development makes several attractive opportunities available. High school students who have had some calculus will be able to verify the part they have studied using the testing mechanisms. Review possibilities are of course at hand. Students can learn those portions of the course that were not covered in high school and complete the first semester of calculus without laboriously repeating the entire course. For instance, students who had some calculus in high school could use the PLATO course to receive credit for the first semester course and prepare themselves to begin the second semester at a university level. As other courses are available, similar episodes could take place.

Since the courses will be modular in nature, they will provide a welcome source of review for students who have difficulty with a certain aspect of calculus later in their studies. An engineering instructor whose students are less than accurate in using the chain rule might refer those students to the lesson and drill on chain rule.

PLATO is used in the courses at different stages as text, as tutorial, as drill, and in testing. The graphic capabilities of PLATO are instrumental in producing high quality courseware. One of the chief advantages of the computer-based instruction is the ability to control the sequencing so that students do respond as the material develops. Questions which in textbooks are purely rhetorical can require a student response before continuation. The touch panel on the monitor screen can be used to ask students to indicate the behavior of a graph of a function near a certain point. Drill exercises with hints and immediate wrong answer responses can be effective learning devices.

# RENTING VS OWNING - COLLEGE LEVEL HOUSING

Mildred C. Luckhardt

East Texas State University

Variable data to be entered:
1. Number of years planning to live in house (25 max.)
2. Marital Status
3. Income (if married both)
4. Total number of dependents (count yourself as 1)
5. Standard deduction used when reporting income (0) or value (1)

(Renting data)
6. Value of Monthly rent
7. Percentage increases in rent annually
8. Monthly cost of utilities
9. Annual insurance
10. Deposit not returned

(Ownership data)
11. Selling price of house
12. Down payment
13. Closing costs of purchase
14. Interest rate on mortgage
15. Term of mortgage
16. Utility costs monthly
17. Condominium fees, if any
18. Rent for lot, if any
19. Annual cost of home owners insurance
20. Annual property tax
21. Percentage of increase in value of the current trend in neighborhood
22. Amount of real estate commission if you sold.
23. Closing costs if you sold.

When data input is complete the program is run and the rent vs owning comparison is displayed.
This program has been used by students in an introductory housing course. Students were unfamiliar with computer use, yet within one period they had all run the program, many more than once and could then explain which decision might be better, renting or owning, under the particular circumstance.
A similar problem in Housing and Society by Lindamood and Hanna, took students two full periods to calculate or even check their hand calculations. Test grades improved and interest was higher than before.

This program would be helpful to the general public or to real estate companies.
Print outs of the program will be displayed.

# CAMI AT JOHNSON COUNTY COMMUNITY COLLEGE

Wayne Mackey, Sue Lindsay and Harold Harp

Johnson County Community College

Computer Assisted Mathematics Instruction (CAMI) is a series of micro-computer (Apple II) programs that complement the regular mathematics curriculum. We will present an overview of the entire system, providing examples of the written material used, demonstrations of the various types of computer programs used and the result of an experiment in which a comparison was made between matched classes, one utilizing CAMI and the other not.

## Documentation

Programs are cataloged by topic for each course, referring to page numbers of appropriate course textbooks, a description of the program, and suggestions for the most effective use. In addition, a list by topic of programs available for each different course with reference to the catalog is prepared every semester, and the list for a particular course is distributed to every instructor teaching that course. We will present samples of the catalogs and topic lists for several courses.

## Computer Programs

1. Demo Programs--The demonstration programs are used by the instructor in the classroom although they also can be and are used by individual students. Primarily graphic in nature, they illustrate various concepts through the use of color, movement, and sound on the computer monitor screen. We will present the Rational Function Demo for an example.

2. Student Use Programs--These programs are designed for student use outside the classroom. One type consists of tutorial programs that explain techniques by text and example and then provide questions and problems with instant response to student answers, including varying amounts of diagnostics. Here, we will present the Conic Section and Absolute Value Inequality Tutorials for examples. The other type of student use program allows the student to use the micro-computer as a computational tool to solve problems that would be impractical without computational assistance. Many of the programs require the student to modify certain program lines in order to complete the whole assignment, thus, providing a small bit of programming experience for the student. We will present Linear Programming to illustrate this type.

## Evaluation

The evaluation of the system and its utilization will be discussed using a transparency of usage figures which demonstrate the popularity of the Math Resource Center. We will also present the results of an experiment conducted last fall to measure the effect of computer-based instruction.

# CAI AT LOS ALAMOS NATIONAL LABORATORY

Mary S. Manteuffel

Los Alamos National Laboratory

Los Alamos National Laboratory has over 7000 employees performing scientific research and development. Continuing education is essential to the vitality of the staff. One way continuing education is provided is through computer-assisted instruction (CAI) via the PLATO system. PLATO at Los Alamos has expanded from one terminal in 1978 to eight terminals in 1981 PLATO was selected as our CAI system for the following reasons: existing courseware, lessons are highly interactive, and good quality graphics, simulation, and data collection capabilities.

PLATO has been used heavily in education of Los Alamos computer users. Los Alamos has one of the most powerful computing facilities in the world and we have developed a PLATO lesson that helps users to understand network complexities. The lesson is useful to both sophisticated and novice programmers, uses graphics to describe the network, requires 15-30 minutes to complete, and required 60 hours of designer and programmer time to write. The commercially available computing courseware on PLATO is, however, of high quality, abundant, and relatively bug free. We therefore have not needed to embark on an extensive development effort to keep our usage levels high.

Problems implementing PLATO at Los Alamos have been few. Unlike many installations, we have few computer anxiety difficulties. Our major problem has been that the PLATO system is separate from our computer network. This requires users to sign-on to a different terminal and system to practice what they learn on PLATO.

# TEACHING DESCRIPTIVE INORGANIC CHEMISTRY IN FIRST YEAR UNIVERSITY USING HIGH RESOLUTION GRAPHICS

John S. Martin - University of Alberta

Descriptive Inorganic Chemistry is a detailed exposition of the characteristic compounds of the common elements, and their reactions, correlated with position in the Periodic Table.

It is the most difficult topic of an Introductory Chemistry course to teach effectively, and it tends to be neglected in first year University courses, owing to the pressure of an overcrowded curriculum. Unfortunately, Introductory Chemistry without Descriptive Chemistry is like a language course with no vocabulary.

I have written, and used in my classes, a group of PLATO lessons which take advantage of the high resolution graphics of this system. A standard Periodic Table display is used in all of them, with a touch square for each element. It is possible to have highly discriminating response, and still have room on the display for instruction and feedback.

Here are three of the new lessons:

1. A set of quizzes, which the student answers by touching an element on the Periodic Table display. In some cases, the answer may be required "blind"...without the element symbol displayed. The quizzes are driven by a dataset. An editor lesson allows an instructor to administer his own questions.

2. An interactive presentation of chemical properties, such as electronegativities, acid-base chemistry, and oxidation numbers. In some cases, touching an element produces a more detailed exposition of its properties, or the display of its electron configuration.

3. A "Guess The Formula" game. The computer tells the student how many elements are in the formula, and gives some hints. The student guesses the identity of the various elements by picking an element off the Periodic Table with his finger and putting it into the appropriate slot. Performance on this is about 80% skill and 20% luck, and is enhanced by knowledge of the properties presented in the previous lesson.

These lessons will all be available on terminals in the University of Alberta display area.

# CAI TUTORIALS IN INTRODUCTORY CHEMISTRY

John S. Martin - University of Alberta

Introductory University Chemistry contains a great deal of material which is best learned by practice. It is usually taught in tutorials, in which a student attempts to solve assigned problems, guided by a teaching assistant.

I have written PLATO lessons which replace conventional tutorials. When complete, there will be 4 to 6 half-hour lessons in each of these topics:

1. Thermochemistry and Bond Energy
2. Chemical Equilibrium
3. Bronsted Acid-Base Equilibrium
4. Redox Reactions and Electrochemistry
5. Chemical Kinetics

The lessons are strongly interactive. The aim is to provide helpful, relevant feedback to every conceivable student response. Some novel judgement procedures have been devised in an attempt to do this.

Each concept is used in examples and exercises as soon as it is presented. The problem banks are large enougn that a student can review several times without encountering the same question again. Mastery criteria vary according to the concepts. In some cases, where total command is sought, the student must get right a certain number of answers in a row, in a timed run. Most assignments are much easier than that!

In the 1981-82 academic year, one semester's worth of CAI material was finished, mostly in topics 1 and 3, and used in two sections of Introductory Chemistry at the University of Alberta. Student response was enthusiastic. CAI was much preferred over conventional tutorials, and there was evidence of improved performance on examinations.

Some of these lessons will be available for demonstration on terminals in the University of Alberta display area.

# ROBUST COURSEWARE DEVELOPMENT

Paul G. Matthews - Bell Laboratories

Bell Laboratories (Piscataway, N.J.) has developed courseware about the UNIX* system and allied software tools using an internally developed CAI (computer-assisted instruction) system. This experience has led to the evolution of courseware requirements, authoring practices, and other standards that might benefit other CAI practioners. Issues relating to courseware organization, alternative uses of courseware components, courseware maintenance and change, and economic constraints are discussed. Courseware about basic UNIX system commands and the UNIX text editor is used to illustrate specific points.

*UNIX is a trademark of Bell Laboratories.

# CAI AT THE UND SCHOOL OF MEDICINE

Mary Fredricks Medal

University of North Dakota School of Medicine

The University of North Dakota School of Medicine faces a unique set of challenges in providing educational services to medical students at four major cities and numerous small communities throughout the state. With a small population and a large geographic area, this task is indeed a difficult one.

To meet this challenge and also to provide exposure of its students to the use of computers, the UND School of Medicine initiated CAI activity in 1979 with the installation of the Interactive Instructional System. In 1980 a full-time Coordinator was hired to assist faculty in the design and implementation of CAI courses and to serve in an advisory capacity to the Administration of the UND School of Medicine in their efforst to extend CAI capability to all four campuses throughout the state.

The Coordinator's activities are under the auspices of the Office of Academic Affairs and the Committee for Computers in Medical Education. Since its formation the Committee for Computers in Medical Education, hereafter referred to as the CCME, has initiated a proposal process through which it reviews all proposed CAI projects. Under its auspices nearly forty hours of CAI software in the areas of physiology, microbiology, pathology, anatomy, pharmacology and family medicine have been developed.

In 1981 the CCME held a retreat of its members for the purpose of making recommendations to the Dean of Academic Affairs for the development of CAI at the School of Medicine for the next five years. The CCME recommended that the following types of CAI programs be developed along with a rank order priority for each of the programs:

1. Linear drill and practice.
2. Tutorials with branching.
3. Simulations.

The addition of three staff positions to the CAI staff, that of educational strategist, instructional programmer and systems programmer was also suggested.

Regarding appropriate hardware and software for CAI, the CCME suggested a combination of the current mainframe system and the use of microcomputers. Recognizing the plethora of microcomputer hardware and software available, the CCME stressed the importance of standardization of hardware and software at the UND School of Medicine.

The recommendations of the CCME have been incorporated into a five-year plan which provides for the extension of CAI services to the three remaining campuses of the School of Medicine in addition to the Grand Forks campus. Some of the problems which we now face with the implementation of this plan are as follows:

1. Stimulation of faculty involvement in CAI activity.
2. Student acceptance of CAI as an alternate mode of instruction.
3. Poor response time with mainframe IIS.
4. Providing a rationale for the addition of microcomputers to our CAI program.
5. Deciding which microcomputer system will best suit our needs.
6. Standardization of microcomputer hardware-software for CAI at the UND School of Medicine.
7. Integration of microcomputer-based CAI with mainframe-based CAI so as to take advantage of the capabilities of each.

# MELODIC DICTATION VIA PLATO: A PROGRESS REPORT

Steven R. Newcomb

Florida State University

Two years ago the Center for Music Research began teaching melodic dictation to all undergraduate music majors at the FSU School of Music via the Gooch Synthetic Woodwind-based MEDICI PLATO courseware. MEDICI was developed for the School of Music by Bradley Weage and Peter Spencer of the Center for Music Research. This progress report deals with some of the unanticipated problems which were overcome in the progression of MEDICI from pipe dream to accredited instruction. Certain competency levels, as measured by MEDICI, are now required of all undergraduate music students at The Florida State University. Each year, approximately 150 new students begin the journey from initiate to certified trained listener.

# COMPUTER-BASED INSTRUCTION PROJECT (CBIP)

Thomas C. Reeves

University of Maryland

The Center for Instructional Development and Evaluation at the University of Maryland University College has a contract through the Army Continuing Education System to develop a military-specific micro-computer/videodisc instructional program in functional literacy. The Computer-Based Instruction Project, ongoing since 1979, has for the 1981-82 contract year two major components.

1. The continued implementation of 85 computer-based functional literacy lessons at ten large and remote army sites in central and southern Germany. These lessons, designed specifically for wheel vehicle mechanics and armor crewmen, are the subject of an intensive evaluation. This evaluation is aimed at systematically documenting the lesson usage, collecting information for improving the lessons, assessing the immediate learner effectivenss of the lessons, judging long range impact of the project, and determining the cost-effectiveness of this approach to instruction.

2. The design, development, demonstration, and evaluation of a microcomputer/videodisc instructional program in functional literacy. This component of the program includes three innovative elements:

a. A new hardware configuration which includes single screen integration of videodisc and computer-generated images, computer-generated sound, touch-sensitive screen, and hard disk storage.

b. An instructional approach based on extensive use of highly motivating simulations, e.g., a time travel simulation in which soldiers journey backward and forward in time to apply functional literacy skills at the Battle of the Bulge and in a futuristic alien city.

c. State-of-the-art functional literacy instruction which emphasizes learning basic skills from a problem-solving perspective in relevant military contexts. For example, the reading component stresses reading for comprehension and application in the common tasks (see, communicate, navigate, shoot, and survive) of all soldiers.

This second component also involves an extensive evaluation effort. One innovative aspect of this evaluation is that, instead of using the multiple-choice pretests and post-tests found in many CBI programs, these lessons employ problems in simulated environments for diagnosis of student needs and assessment of immediate effectiveness. For example, a soldier is called upon to read a WWII Stars and Stripes to determine the time and place in which his/her errant time machine has placed the soldier. If the soldier successfully identifies the time and location, he/she proceeds in the simulation. If not, the soldier is immediately presented with instruction matched to the required functional literacy skills.

# FRENCH UNSTABLE e

Bernard Rochet and Jim Horne

University of Alberta

The purpose of the lesson is to present a thorough and systematic description of the behaviour of unstable e in Modern Standard French, taking into account the most recent research reports on the topic. Although the nature of the question requires a failry complex treatment, the lesson is designed so that it can be used by students with little prior training in French.

CAL was selected for the following reasons: It is well suited to the presentation of the behaviour of unstable e by means of an algorithm; it allows for extensive and varied feedback - a necessity in view of the numerous factors involved; it makes it possible for students with widely different backgrounds in French to use the lesson as efficiently as possible, by allowing the more advanced ones to skip the first part of the lesson ("Identification of unstable e") and to proceed directly to the deletion/retention rules after demonstrating their level of competence through an optional pre-test.

A number of features were incorporated to make the lesson as flexible and as informative as possible: Lists of exceptional items are contained in HELP sequences and can be obtained from any point in the lesson; a historical note explaining the origin of a puzzling irregularity is also available for the more inquisitive students, but is not part of the main body of the lesson; stylistic variation is handled in quizzes by means of double-barreled feed-back and through a conversational exchange with the student.

On the programming side, while simple cases are handled by yes-no questions and answers (typed in or given on the touch panel), longer utterances containing several tokens of unstable e required a more complex approach with moveable arrows and multi-step feed-back. A technique of using interactive flow-charts proved to be very successful.

A student questionnaire showed that students approved overwhelmingly of PLATO as a teaching medium.

# THE USE OF COMPUTER AIDED INSTRUCTION IN THE NUCLEAR INDUSTRY

D. L. Rogosky

Westinghouse Electric Corporation

Westinghouse Electric Corporation has offered nuclear training programs since 1971 at the Westinghouse Nuclear Training Center, located in Zion, Illinois. The Training Center offers complete programs in initial license training and periodic retraining. As operator training programs continued to mature at WNTC, other training programs were developed and expanded in the Pittsburgh area. In 1977, the Nuclear Training Services Department of Westinghouse (WNTS) in Pittsburgh, Pennsylvania, began developing specialty training for engineers; comprehensive training for instrumentation and control technicians, on-site plant specific training for operators, technicians, and engineers; plant specific programs for senior reactor operator and reactor operator license; and curriculum materials support services for utilities conducting their own training programs. An Emergency Technical Center is in progress at both WNTC and WNTS to assist plant operators during an emergency.

With one decade of formal training and thousands of man-years of experience, Westinghouse Nuclear Training Services Department continues to grow through expansion and development of new and improved methods of teaching such as the development and use of a Plant Simulator and the integration of Computer Aided Instruction (PLATO) in NTS courses.

NTS is responsible for nuclear plant personnel training, development of plant specific training materials, and training management services to utility customers. One method of teaching nuclear plant personnel is on a plant simulator. These simulators are replicas of an operator's station, a simplified generic control panel or a general purpose video display unit and keyboard. Mathematical simulations of complex engineering systems are developed and then used to interface with the learner through either a special control panel or a CRT and keyboard. Utilizing this method the students are faced with hypothetical situations which provide realistic hands-on experience without jeopardizing the safety of the people in the nuclear power plant.

A more recent method in progress at WNTC and WNTS is the development of materials on a computer system (PLATO) to aid the instructors in their teaching. Over the past two years WNTS has expanded our development efforts to investigate many different types of Computer Aided Instruction Systems and also to train six instructors in the use of Control Data Corporation's PLATO system. During this time ten PLATO terminals have been purchased for courseware development and student training. Small amounts of material for trial use have been generated on PLATO by our instructors. WNTS is beginning to work with consultants from the University of Delaware to develop new material for the system.

Another step WNTS has taken is the commissioning of Penn State University to write a lesson using interactive graphics, starting with a mathematical mode of the PWR pressurizer. We expect to develop one classroom equipped with audio visual aids and various computer terminals for the instructor and his students to use.

We feel Computer Aided Instruction is an effective learning tool in our training programs and we anticipate that it will grow into an important and strong contributor to the quality and attractiveness of our training programs.

# USING PLATO TO TEACH LIBRARY RESEARCH SKILLS

Judith Sandler, Patricia Arnott, Patricia Fitzgerald

University of Delaware

Basic library research skills are taught to all University of Delaware freshmen as a component of an English department course in "Critical Reading and Writing". Computer-based instruction was chosen as a replacement for the traditional lectures on this subject by reference librarians because of its advantages to the librarians, students and English department instructors. Librarians now use the seventy-five hours per semester previously spent lecturing to provide a more complex and individualized level of bibliographic instruction. Students are able to learn at their own pace; get feedback on concept acquisition; and see examples from indexes and catalog cards which were not easily visualized in traditional lectures. Instructors receive a record of the student's progress on PLATO.

The library research lessons are written in tutorial mode with built in drill and practice. After completing the lessons, there is a forty question on-line test that covers all five lessons. Each year approximately three thousand University of Delaware freshmen will use these lessons to develop library search strategies.

The five lessons cover the card catalog, periodical indexes, newspaper indexes, government documents, and locating references in the University of Delaware Library. In the first lesson students see examples of title, author, and subject cards, and identify parts of a catalog card. The Library of Congress Subjects Headings is introduced as an aid to finding usable subject headings in the card catalog. Students using the next three lessons interpret information from both periodical and newspaper index citations, and from entries in the monthly index to government publications. Students also identify the elements of a serial records card and a holding card. The last lesson in the series presents information on the physical location of books, periodicals, newspapers, and government documents in the University of Delaware Library.

The lessons were designed by a team of three University of Delaware reference librarians with support from the Office of Computer-Based Instruction. They were programmed during the spring of 1981 and have been used by twelve hundred students since then. During the 1981 fall semester approximately one-third of the freshmen students were taught library research methods in a traditional lecture, while the other two-thirds used the PLATO lessons. Data are still being analyzed, but preliminary results of pre- and post-tests show that the lecture and the PLATO groups learned equally well. Because of the large number of students involved, it is hoped that data obtained from the use of these lessons will not only improve library instruction, but will also provide information that will improve CAI instruction in general.

# DELIVERY OF CONTINUING MEDICAL EDUCATION BY COMPUTER ASSISTED INSTRUCTION

E. N. Skakun, M. W. Petruk, C. Park, R. S. McLaughlin

University of Alberta

Continuing medical education (CME) which is the longest phase of a physician's education is a complex activity. The bulk of formal CME activities is comprised of classical instruction including courses, seminars, and workshops. During the past decade, the effectiveness of computer assisted instruction (CAI) as an educational tool has been demonstrated. In response to this, commercial firms have recognized this potential and have developed CME materials which can be used in conjunction with computer technology. One such commercial firm is the Milliken Communications Corporation who have developed a series of computer-based medical seminars for presentation on the APLE micro-computer.

The purpose of this project is to determine the feasibility of using the Milliken computer-based material for CME. More specifically the project assesses the APPLE micro-computer focusing on aspects such as adequacy of instructions to use the terminal, ease of using the equipment, and appropriateness of using the computer for presenting CME material. A second component assesses the content of the Milliken prepared seminars by focusing on aspects dealing with appropriateness, relevance, educational-worth, and accuracy. Questionnaires relating to these aspects have been constructed.

Preliminary results based on a small number of practicing physicians who have participated in the project indicate that no major difficulties are encountered in using the APPLE system and that the content of the seminars is relevant and appropriate.

# DEVELOPMENT OF MICROCOMPUTER-BASED INSTRUCTIONAL SYSTEMS

Nick Van Matre

Navy Personnel Research & Development Center

The Navy Personnel Research and Development Center, San Diego has initiated a new project to develop a training system based on microcomputers that combines the technologies of Functional Context Training (FCT), mastery instruction a la Dr. Fred Keller's Personalized System of Instruction (PSI) and graphic simulations of equipment and job processes.

The heart of this instructional approach (FCT) structures training in a highly job-oriented manner with fundamentals and theory being taught only when necessary for the student to continue learning how to do the job. Additionally, familiarization with equipment is done in a "whole to part" manner, with the student receiving instruction only on a task or operation that is necessary for the student to continue learning the job. Although FCT was devised over a decade ago, it has not been extensively implemented due to its reliance on expensive actual equipment. The technological development of microcomputers along with advances in simulation have enabled such a training approach to be attempted without all of the real equipment. By using mastery, individualized instruction additional savings may be possible through a reduction in training time.

The approach to be taken with this research is to develop the instructional management system, develop prototype instructional materials for a small portion of an operational Navy technical training course, and if those are successful, develop a major portion of the course materials using FCT principles. The evaluation of the project will include measures of on the job performance in addition to the conventional end of training assessment procedures. Bottom-line evaluation will be conducted through a cost-analysis comparing conventional with FCT instruction using microcomputers.

# CHARON: A CURRICULUM MANAGEMENT SYSTEM BASED ON INSTRUCTIONAL GROUPS

R. L. Venezky, G. A. Feurer, K. Kahn

University of Delaware

CHARON is a PLATO program that allows school districts to establish instructional calendars for curricular subjects (e.g., reading, math), to generate schedules based upon these calendars, and to gather cumulative statistics on performance of instructional groups according to test mastery and content covered relative to expected coverage.

CHARON has two particular goals. The first is to allow a classroom teacher to designate the teaching materials he or she wants to use for the year and then have these materials mapped onto an instructional calendar. This establishes an instructional pace for the year, indicating which lessons should be covered on each instructional day. The second goal is to allow principals, reading specialists, and district level personnel to monitor reading achievement through cumulative reports that show where students are relative to where they are expected to be in the school calendar. The system also allows for test score data to be entered and summarized across classrooms, schools, or grade levels within the total school district.

The program has several different modules. One module keeps information on all the materials that are used for instruction within a particular subject. Materials are defined hierarchically with the material itself as the root and lessons as leaves. Two subdivisions between these levels are allowed. Lessons are then defined according to the amount of time required to complete them, based on slow, average, and fast paces.

The second part of the system holds the instructional calendar for the year. This indicates which days during the school year are available for instruction. Another module carries information on the district schools. Each school is identified by name and principal, along with its teachers and their instructional groups for the particular curriculum that is being monitored. Each instructional group is then assigned the materials that they should cover for the year. A teacher can designate where instruction will begin and end within any particular text, and the pace that is to be followed (slow, average, fast). From this information the system generates an instructional calendar for each instructional group.

At designated checkpoints during the year, teachers report where they actually are in the curriculum, along with the last test scores they obtained. The system then prints summary reports as mentioned above, both for the number of days ahead or behind according to the schedule, and for the mastery and non-mastery data gathered from the various classroom reports.

CHARON is written in TUTOR for the PLATO system at the University of Delaware, and is currently being used by four elementary schools in Philadelphia as part of a pilot project to improve district reading instruction.

# SIGHT-SINGING VIA PLATO: A PROGRESS REPORT

Bradley K. Weage - Florida State University

Several years ago at a Music Educator's National Conference in Chicago, Dr. Bruce Benward proposed that a device be built which would make it possible to teach sight-singing via computer. At that time, such an application was generally not thought feasible. Much later, using research funds provided by the Center for Music Research, Sherwin Gooch designed a PLATO terminal peripheral capable of extracting musical pitch of sufficient resolution, and at a rate sufficient to teach sight singing. This digital pitch extractor was later implemented at the Computer-based Education Research Laboratory (C.E.R.L.) at the University of Illinois by Gary Czvitkovicz of the C.E.R.L. Music Group, under the direction of Lippold Haken.

Assembly language control programs for both the Pitch Extractor (PE) and the PLATO terminal have been completed, as well as TUTOR utility routines which instantaneously convert PE-derived information into standard musical notation. Sight-singing courseware is currently in development.

The PE is on display in the exhibit area. Sight-singing via computer is nearly a reality.

# REPUBLIC VII: A PORTABLE MUSIC STATION FOR USE WITH A REMOTE HOST COMPUTER LIBRARY

Bradley K. Weage - Florida State University

An architecture has been chosen for a music station to be used with the Integrated Cybernetic Music System (ICMS). A prototype, based on the I.E.E.E. S-100 bus has been constructed. Many optional hardware components may be used, including high resolution color graphic display, Winchester hard disk, Z-8000 processor, and music peripherals. Software packages which allow the Republic VII computation station to be used as a full color PLATO terminal, a TELEX (or other standard ASCII) terminal, or a stand-alone processor have been written, converted, or interfaced.